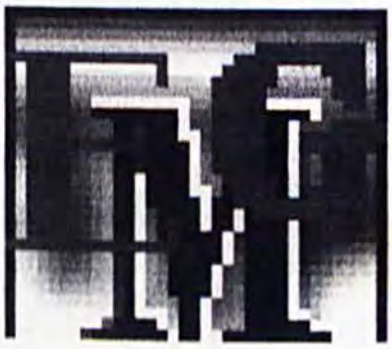
EDUCATIONAL MATHEMATICS GAMES (EMG) FOR ALGEBRA

WXES3182

Prepared by: Liew Yuh Tian
(WEK990099)

Supervisor: Prof. Madya Raja Noor Ainon Zabariah Raja Zainal Abidin

Moderator: Mr. Mohd. Khalit Othman



ABSTRACT

As one of the undergraduates pursuing Degree in Bachelor of Computer Science at Computer Science and Information Technology Faculty (FSKTM), University of Malaya, are required to carry out a thesis project as a partial fulfilment of the requirements for the course.

Thus, my project title is **Educational Mathematics Games (EMG) For Algebra**. It is an educational computer games system. With the motive of introducing algebra knowledge to the students more effectively, the system implements the concept of multimedia, which integrates text, graphics, animations and audio into the games' display, to make the games more attractive especially to the teenagers.

EMG For Algebra emphasize on the addition, subtraction, multiplication and division operations unto algebra, and also the polynomial as well. The EMG of Algebra covers largely portion of secondary schools' mathematics syllabus for algebra. Therefore, the main target users of EMG for Algebra are secondary school students, their teachers and parents. The games will be presented in the forms of multiple choices question, subjective question, 'falling bricks' and puzzle.

Visual Basic 6.0 was used in system prototyping and coding process. Microsoft Access 2000 was used to create database. While Adobe Photoshop 6.0 and MorphStudio Image Editor was used to edit the graphics for the system. Mp3Trim and Mp3 To Wave Converter used to edit the sound implemented in the system

Finally, I hope the EMG for Algebra can serve to enable players enjoy themselves throughout the playing and learning process, and to achieve the objective specified before the system is developed.

ACKNOWLEDGEMENT

How peace I am, for the pass two semesters, with the help from so many individuals, I get my system developed. Thank you everybody.

First of all, I would like to express my utmost gratitude to Professor Madya Puan Raja Noor Ainon, my thesis project supervisor and Mr. Mohd. Khalit Othman, the moderator of my project, who had guided me along the way to come out with my games system. They gave me their comments, suggestions and advices in the interface design, games environment monitoring and so on in the pass few weeks.

Special thanks to the lecturers, tutors, other academic and technical support staffs of Computer Science and Information Technology Faculty (FSKTM), University of Malaya, especially to those who had assisted me when I was doing my literature review and system coding. For those who had given me their advices and support when I was preparing for VIVA and compiling thesis report, thank you very much.

I would also like to thank my course mates in the faculty, seniors, housemates in Petaling Jaya and all my friends who had made my work easier from time to time. They had also made my days more brightly and colourful, especially my course mates, we were helping and motivating each other from time to time, and exchange information as well.

Last but not least, I would like to express my gratitude to my family members for their support and understanding for the past almost a year. Wish everyone feel good, may all be well and happy. Thank you.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Project Overview

Educational Mathematics Games (EMG) For Algebra is an educational computer game. It integrates **numbers, text, graphics, animations and audio** to introduce mathematical knowledge to the target users in a more advance and effective way. The concept of multimedia is implemented into the games. This is to make the games more attractive especially to the teenagers.

The EMG For Algebra consists of 3 chapters;

1. + (add) and – (subtract) mathematical operation for algebra
   - $4p + 7s = 68$
2. x (multiply) and / (divide) mathematical operation for algebra
   - $(4ps) / (6p) = 33$
3. Polynomial
   - $3p^2 + 7p - 10 = 0$

Each chapter consists of 5 modules;

1. Multiple choice (Objective) question
2. Subjective question
3. Falling bricks
4. Puzzle
5. Exit chapter

Each module (exclude the 5^{th} module) then has 3 levels of difficulty.

Player(s) will get their reward (marks or keys) when

1. Answer a question correctly within time limit if there is (add marks).
2. Answer series of questions correctly (add marks and bonus).

3. Total marks collected more than predetermined amount. (add a key)

Otherwise, part of their rewards will be removed if they not able to answer the question correctly. The correct answer will be shown immediately after the time limit is up or the player(s) submit the wrong answer.

Besides, 'Help on EMG For Algebra' and 'Exit Game' functions are provided to assist user(s) whenever necessary.

1.2 Project Scopes and Target Users

The project scopes and target users are as follows;

SCOPES

EMG For Algebra covers the basic part of the world of algebra. The game emphasizes on the

- \+ (add),
- \- (subtract),
- x (multiply),
- / (divide) operations unto algebra and
- n^2 polynomial

This game covers largely portion of secondary schools' mathematics syllabus (Form1 to Form 5) for algebra.

TARGET USERS

1. Secondary school students (Form 1 to Form 5).
2. Pre-university (Form 6) and college students.
3. The teachers who teach those students mentioned in (1) and intend to motivate their student in the process of learning algebra.

4. The parents who have some basic knowledge about algebra and intend to motivate their children in the process of learning algebra.
5. Anybody else who are interested in this EMG or wish to learn algebra through playing computer games.

1.3 Motivation

The main reason I would like to develop an EMG by using computer and multimedia technologies is to change most people's perception towards the process of learning mathematics. The mathematics learning process is no longer bored but full of fun and challenging if we can put in the element of multimedia and games.

Secondly, the parent age generation always has a negative view towards computer games. They take playing computer games as an activity that causes the waste of time, energy and money, which good boys and good girls should never touch it.

I would like to eradicate this wrong perception by introduce this EMG, the parents are encouraged to play EMG together with their children. I don't see the point parents and teachers will go against this because it is an EDUCATIONAL game!

Today, the computer has become an essential part of the world and greatly influences the field of entertainment and education. Varies types of educational computer games had emerged consequents of that, EMG For Algebra is one of the example. Undoubtedly, the computer technology helps students to learn at their own pace and up to their full potential, while multimedia elements make learning processes more interesting. Thus, educational sector can no longer neglect the mass of computer and multimedia technologies, now and after.

1.4 Project Objectives

The main objectives of developing this Educational Mathematics Games (EMG) For Algebra are as below:

1. To enable the players enjoy themselves throughout the learning process by playing EMG For Algebra computer games. The algebra topic will no longer be bored and dull but full of fun and challenging.
2. To train the players think fast and have confident with their own answers. The players are much encouraged to come out with their own 'shortcut' way (any other alternative way) of solution methods, rather than just memorize the textbooks all the time.
3. To implant the latest technology (computer aided and multimedia) into the mathematics learning process. To enable players learn on their own rate and up to their full potential.

1.5 About Educational Mathematics Games (EMG) For Algebra

EMG For Algebra consists of 3 chapters and each chapter consists of 5 modules. Each module (exclude the 5^{th} module) then has 3 levels of difficulty. This is to make the games more challenging and to suite different standards of players. The different level of difficulty is distinguished by the given time limit and the form of the question.

Further information for the five modules

1. Objective question: A question is followed by several choices, the player is required to pick the correct answer.
2. Subjective question: A question is followed by blank answer box, the player is required to type in the correct answer.
3. Falling bricks: Some bricks (each contains answer) falling from the upper part of the screen, the wall (contains questions) is remained static at the bottom part of the screen. The player(s) is required to fix the proper brick to the wall by matching the questions and answers, before the brick(s) touches the wall.

4. Puzzle: Puzzle pieces (each contains number or algebra, or mathematical operator) are placed surrounding the puzzle board (contains mathematical equations and a few blank squares), the player are required to place the puzzle pieces into the board (place into the blank squares) correctly.
5. Exit chapter: Enable users quit from the chapter and back to the main menu

1.6 Goals and Limits

Goals

- Students begin to like mathematics because it is full of fun and challenging.
- More made in Malaysia EMG products emerge.
- More Malaysians involve in scientific and mathematics research area.

Limits

- *Time factor*. This project has to be finished within 2 semesters, therefore it is quite impossible for me to go in very detail research and prototyping work when developing the games. Moreover, I have to do it solo.
- *Only covers secondary school syllabus*. This game covers only basic part of the world of algebra and does not cater for those undergraduates. Even thought the form 6 and college students, which are listed as target users, cannot really depend on the EMG For Algebra because mostly questions are too simple for them.
- *Limited question types*. Only 4 question types, which are multiple choice questions, subjective questions, falling bricks and puzzle.
- *Limited number of question*. Due to the first limit – time factor that mentioned above, there are limited variety of questions in the databases. Player(s) could have getting bored after playing many rounds in the same module because the questions keep on repeating.

1.7 System Requirements

The computer system requirements recommended to run this game are as below:

1. System using 500 MHz (or above) processor
2. 128MB of primary memory space
3. 20MB (or more) of available hard disk space

The environments required to developed the game are:

1. Windows 98 or Windows 2000 operating system
2. Microsoft Visual Basic 6.0
3. Microsoft Access 2000
4. Adobe Photoshop 6.0
5. MorphStudio Image Editor
6. mp3Trim
7. Mp3 To Wave Converter

1.8 Report Organization

Chapter 1 Introduction

This chapter serves as an introduction to the entire report. The project overview, scopes and target users, motivation, project objectives, goals and limits and the brief introduction of EMG For Algebra and its nature are included in this chapter.

Chapter 2 Literature Review

This chapter records the research done and some information for project development process. It mainly consists of discussion on what EMG is, following by a review and comparison on several EMG packages that existing today. This chapter also contains some information on the planning and development of a project, the capabilities and components of multimedia application, various authoring tools and eventually the reviews and comparison of other computer games which existing today.

Chapter 3 Methodology

This chapter discusses on the importance and the flow of project management and project development strategy. This including varies project development life cycle and software prototyping methodology. The chapter will emphasize on Waterfall software development life cycle model, which the chosen SDLC model for developing the EMG For Algebra. The project schedule is also shown in this chapter.

Chapter 4 System Analysis

This chapter describes the initial system development phase – system analysis. This chapter records the reasons of choosing those chosen development tools, which will be used to compose the interactive user interface and to develop the system. The synthesis on other reviewed systems and the proposed methodology to develop EMG For Algebra. Besides, the functional and non-functional requirement analysis, the software and hardware requirements will be verified and discussed in this chapter.

Chapter 5 System Design

This chapter documents the system design process. The pyramid analysis diagram is used to figure out the hierarchical dimension if the system. This system is divided into chapters and modules and accomplish by graphical user interface screen display design. The database design and screen design are planned in this chapter. Besides, a few lines discuss on the project outcome will also recorded at the ending part of this chapter.

Chapter 6 System Development and Implementation

This section will show some of the minor modifications made to the previous design during the system development process.

The important part of the Visual BASIC 6.0 coding is shown for the purpose of further explanation.

Chapter 7 System Testing

This chapter discusses about the purpose of system testing and testing strategy. In addition, this chapter will discuss in detail about the unit testing strategies, integration testing and system testing focus, which applied to the EMG for Algebra.

Chapter 8 System Evaluation and Conclusion

This chapter will list down the system strengths and its limitations, the future enhancements for EMG for Algebra, problems encountered when developing the system and the solutions. Here, the further enhancements for the system are also suggested. Finally, the conclusion of this project will be included at the end of this chapter.

CHAPTER 2: LITERATURE REVIEW

2.1 Educational Mathematics Games (EMG) – Overview

Computers have the capacity to provide a broad range of rich mathematical experiences. With computers, for instance, children can use and develop their skills in geometry, data, algebra, and computation as they solve engaging puzzles and problems, play adventure games, and create on-screen machines, designs, puzzles, and even 'cities'. Computers can also offer players opportunities to work at an appropriate level of challenge, and to advance to more difficult levels in the future.

Despite the potential of computers to engage us in significant mathematics learning, many mathematical computer games focus only on drill of number facts and practice with computation. There is a great deal more to fundamental mathematics for students; and, there is a great deal more that computers can offer to help students with this content.

The main point of build an EMG is to engage the players in appealing mathematical activities, they come to appreciate that math is useful and interesting in itself: exploring math can be the central "play" of a computer game.

To Choose A Good EMG

When evaluating the mathematical content of an EMG, consider the following:

1. What math does a player need to know in order to begin playing?
2. What math will players learn?
3. What role does math play in the game?
4. Will players enjoy and appreciate the math in the game?
5. Does the game offer levels of difficulty?

Is The EMG Equitable?

It is important to engage students in using EMGs by offering them experiences that are both enjoyable and educational, rather than by playing to inaccurate and potentially

harmful stereotypes. Many computer games have been created involve a great deal of conflict and violence, primarily male characters, and competition. In recent years, some computer game designers have attempted to attract girls to computers by offering games that focus on themes such as shopping, dating, and make-up. However, for the most part, these efforts have not yet resulted in quality educational games software.

Another aspect to equity is learning style: some learners enjoy being presented with problems, others like to pose their own problems; some like to design and invent, others prefer to extend and adapt what others have created; some are motivated by time pressure, others find it distressing. Computers have the potential to address the needs of a wide variety of learning styles, and sometimes many needs can be accommodated within a single program.

When evaluating whether an EMG is equitable, consider the following:

1. Who are the main characters in the game?
2. What is the context or story of the game?
3. Are there possibilities for working collaboratively?
4. Does the game contain any harmful gender, ethnic, or other stereotypes?
5. What kind of learning environment does the game offer (e.g. time pressure, opportunities to design and invent, solving given problems)?

How To Identify A Good EMG?

A good game, whether the EMGs or else games, is one that players want to play again and again. There are several characteristics that good games share: the goal and rules are clear; it's easy for players to keep track of their progress as they play; the game can be played with a variety of strategies; the game offers variety (for instance, because players can make different choices, or the game contains a random element such as a die); and the game is so motivating that players are willing to persevere when facing challenges and to work to improve their strategies so they can become better players. Although many educational computer games available today offer attention-getting graphics,

sound, and other special effects, these can become tiresome if the game itself is not well-structured and appropriately challenging.

When evaluating whether a piece of software is a good game, consider the following:

1. Does the game offer variety, so that players can play repeatedly without getting bored?
2. Does the game offer challenge, so that players can continue playing as their skills increase?
3. Does the game have a goal? What is it?
4. How many players can play at once? Do players collaborate or compete?
5. Will the context, characters, and story be appealing to the players?

The Credits Should Go To EMG

It's a computer game: Today, the educationalists have found out one that traditional education largely ignores, which is learning through playing games. Many students find mathematics a boring subject, a lot of formulas to be memorized, a lot of theorems to be applied in appropriate situations, which most of the formulas and theorems have very little connections to their daily life. Many students by their own nature like to play games rather than solving mathematics problems. Some of them even visit arcade centers, which they are not suppose to enter, to play computer games. Thanks to the first person who were so innovative to put mathematics and computer games in the crossroad and came out with the first EMG (educational mathematics games). Very few students feel reluctant to learn mathematics through playing games. Nowadays, students can enjoy their learning process by merging their hobby and learning in a way.

On the spot evaluation: With the advance of computer hardware and software products, the processing speed and memory capacity has increased. The computers are really so powerful to deliver first-handed learning experiences. The respond time of computers is becoming shorter and shorter, this causes our PCs be able to interact with users in real time. Therefore, almost all of the EMGs in the market provide real time respond to the

players. This is a very important factor to consider when we are trying to use EMGs as learning aid tools. When players answer a question correctly or make a correct movement, reward shall be given immediately to motivate them to keep on exploring and learning. Similarly if players make false movements, the system must be able to give proper respond immediately, it can be sort of punishment like minus marks, a warning prompt out or else, then the system must be able to show proper steps and answers if the players wish to know. From here the players can learn from mistakes and improve their problem solving and decision-making skills. Thus, EMGs are better than traditional learning methods in the aspect of the students can be evaluated from time to time without much human intervention.

Provoke innovative thinking: The creation of EMG itself is already an innovative piece of work, many of the games are actually founded by stories, fairy tales, and some other very attractive themes. This is much different from the school textbooks that just with solid facts and theorems. In the games, students are always encouraged to think divergently to solve problems. The mathematical problems are manipulated in many ways to test the students' understanding of the basic theory. Thus, students must explore and try whatever steps they can think off on their own, including those methods never taught in the school.

EMGs for Algebra – Overview

We just discuss about generally how EMGs look like, now I am going to narrow down the scope, which I would like to touch on the EMGs particularly for Algebra.

Very briefly about algebra and its history: Al'Khwarizmi, whose full name is Abu Abd-Allah ibn Musa al'Khwarizmi, was born about AD 790 near Baghdad, and died about AD 850. His most important contribution, written in AD 830, was Hisab al-jabr w' al-muqabala. From the al-jabr in the title we get algebra. The treatise develops a system

for the solutions of quadratic expressions including geometric principles for completing the square.

Al'Khwarizmi also gave a classification system for quadratics. He devotes a chapter to each chapter in his treatise and gives methods in solving each differently.

Six Types of Quadratics

1. Squares equal to roots (x^2 = square root of 2)
2. Squares equal to numbers ($x^2 = 2$)
3. Roots equal to numbers (square root of x = 2)
4. Squares and roots equal to numbers ($x^2 + 3x = 25$)
5. Squares and numbers equal to roots ($x^2 + 1 = 9$)
6. Roots and numbers equal to squares ($3x + 4 = x^2$)

Al'Khwarizmi also gives geometric proofs of these methods. In 1145, Savasorda published Liber embadorum that gave the complete solution of the quadratic equations. In 1494, the first edition of Summa de arithmetica, geometrica, proportioni et proportionalita was published. It sets out equation systematically and algebraically, as in the following translated equations:

(6 + square root of 10)(18 - square root of 90) = (108 - square root of 3240 + square root of 3240 - square root of 900), which is 78

Scipione dal Ferro is the first credited with solving cubic equations algebraically, around 1515. However, he could only solve cubic equations with the form

$x^3 + mx = n$

He kept this work a secret until 1526 when he revealed it to his student Antonio Fior. Soon, the work was common knowledge around Bologna, where dal Ferro taught at the University of Bologna.

Other observations in the field of complex equations were also made, primarily that of Harriot. He observed that if

$x = b, x = c, x = d$ then

$(x - b)(x - c)(x - d) = 0$

which allowed more uses for the cubic equations. Many proofs after this followed, including ones which first proved these principles algebraically, instead of geometrically. The further use of algebra supplemented modern mathematics in a very important way.

The computer technologies went in: Decades ago, the concept of putting algebra into computer games, once esoteric, is now becoming very common in the educational establishments. The whole world waited until near 1980s, group of genius came out with the idea of educational courseware, slowly pushed the educational mathematics games software to be popularize around the world during 1990s until today, of course the EMG For Algebra is one type of EMG products.

The Contribution of Computer Technologies To Learning Process

1. The students can have their own learning styles and preferences, without much interfering from the fixed syllabus, classmates and teachers.
2. The students are free to learn at their own pace and up to their full potential, without much intervention from the school syllabus.
3. Multimedia: more interesting and reality to the students, compare to the facts, theories and theorems which have no connections to their daily life.
4. With the visual, sound and animation effects, it can enlarge one's memory capacity because the students pay more attentions to it.
5. Self-motivation. There is nobody at the side to push you. For example, when the students are attracted to the EMG, they will be very motivated to take the challenge and try to overcome whatever problems encountered.
6. Students develop various intellectual skills and research spirits, by the help of learning aid tools. Thus build up a strong relationship between learners and knowledge.
7. The system can track the students' progress from time to time and recommend next sequence of complexity of task to them.

2.2 EMG - Planning and Development

Certainly I would not discuss very deeply about how to develop an EMG For Algebra in this chapter, but I will state some of the reasons why proper planning before develop a system or a project is so important. A structured approach throughout the planning, and an excellent planning throughout the project development can help to save a great deal of time and resources. Ideally, everything included the resources available, skills or technologies available and the goal of the project have to be identified and planned before the programmers start to code.

There are few factors that must consider and plan before a development process start. Those arguments not only applicable to the EMG For Algebra Project, but also applicable to mostly software development projects.

Overview, Motives and Objectives

First of all, the project overview, the motivations to develop the project and the objectives of the project must be ruled out before any further planning can proceed. The motives to start a project usually come from a person of a group of people, which has strong feeling that something has to be done to fulfill the lacking part of an environment or the society. With the resources and technologies available during the time frame, the particular person or the group of people may come out with ideas or proposals to start a project.

The project's objectives then identified and agreed. Base on that, the project overview, with a rough description of how the user interface will appear, what the project doing and dealing with and the content of the project can be planned.

Resources and Technologies Available

After knowing the outline of the project's content and user interface, one can figure out the resources and technologies needed to support the development of the project. The word 'resources' here means the material such as images, video, graphics, photos, sound, text, animations, hardware and software needed, and how much hour (how long it will take) and money needed.

The technologies available means how advance the current software and hardware are, and nevertheless the most important – how many system analyst, programmers or experts do we have?

With the collected information, one should produce a master plan or a workflow dependent on the resources in hand and the time limit.

Flowcharts

This is sort of diagram expanded from the project outline produced previously. There are ways to draw flowcharts for a project. Basically, all the flowcharts are to serve the purpose of showing the project's main structure to the programmer before any coding process take place.

The flowcharts must take care of as much as the possibility might occur when the user use the system. The user must always given a chance to redo the steps that he/she has done it wrongly, and the system must be able to recover from any failure. The programmer will then code the requirements into the system.

Prototyping

Prototyping is a technique for building a quick and rough version of a desired system or parts of that system. The prototype illustrates the system to the users and the designers. It allows the users and the designers to see flaws and invent ways to improve the system. It serves as a communications vehicle for allowing persons who require the system to review the proposed user interaction with the system.

- A prototype is used where the functions and detailed design of a system are not yet fully understood.
- The prototype is used to explore and solidify the functions and design.

It has been said that 'a picture is worth a thousand words', but a prototype would worth ten thousand words or more. A prototype is fundamentally different from any paper description, it is real and manipulatable. A prototype can be just modified by the

designers as well as the users, thus let the users get a feel for how their system will look like before system delivery.

When a prototype is reviewed seriously by end users, they almost always change something. This implies that, without proper prototype or no prototype at all, a system would have been built that was less than satisfactory. Prototyping does so much to solve the problem of inadequate communication between designers and users.

There are tools in the market today to help the prototyping process. One of them is RAD (Rapid Application Development), the benefits of using RAD is fast and in RAD lifecycle, the prototyping tool should be built with the final development tool so that they pass directly from the user design phase to the construction phase. Other prototyping tools are JAD, SAA, I-CASE Toolset and so on.

Programming and Testing

After everyone has cleared about the system's requirement and characteristic, the project team then can decide which authoring tool, database design tool and multimedia software to use to develop the system. This is very important because if proper tools and software are chosen, great deal of time can be save throughout the development process.

When everything is planned and understood by everybody who are involved, the coding process then can proceed to create the application. The user interface then can be seen.

Since the concept of prototyping is being implemented, the system will be reviewed and tested (module by module at the beginning) from time to time to improve the system. After the system finish developed or almost developed, the programmer will do the initial testing. Once the programmer feels satisfy with the system, then the project is said ended the development part. Further testing process then should be given to the user or other parties. The feedback from there should be consider and try to modify the system if necessary.

Evaluation

Finally, come to the evaluation part. Sometimes, it can be included into the prototyping process but sometimes this is impossible. For an educational computer games or any other educational courseware, the best target audiences to test with are the students, their parents and their teachers. So, one can start to test it with a group of students, let them have their time to play around with the games, maybe for a few days or weeks. They might be interviewed everyday or at the last day of the testing period, the feedback from them should then be reviewed.

The are ways to get the feedback from them who involved in the evaluation process, one is interview but we still can provide questionnaires, and any information that they wish to give us through any other ways. All information obtained is very useful for us to improve the system in the future.

2.3 Multimedia

Consequents from the mass and advance of science and technology especially the computer technology, the concept of multimedia was widely spread after 1980 and today it is almost apply to every field in our society. The sectors that most influenced are computing sector, telecommunicating sector, publishing sector, education sector, entertainment sector and television/radio broadcasting sector. The advance of multimedia technology today has a great impact unto the entertainment sector especially to the making of movies.

Before the computer multimedia comes in, there were slides projection with human narration and movies that brought up the points of multimedia. Both the early time multimedia tools had already played very important roles in the education sector. However, nobody use the term multimedia at that time.

Hence, 'multimedia' is quite a new word and new term in English, many professionals found it difficult to give it a definition that can satisfy every context.

Definition

Multimedia is one of the most overused term in our society today. The word 'multimedia' only came the world around the year of 1990, at the beginning of the 1990s, many people even the professionals could not agree with each other about the definition of 'multimedia', the worst was went until, "multimedia is undefined". Even though until today, the situation still goes, "if you asked 10 different people for the definition of multimedia, you will certainly get at least 10 different answers". This is partly because the concept of multimedia is much depending on the environments, and the fields that it applied.

However, in our context, 'multimedia' can be derived this way:

"Multimedia is a combination of images, text, graphics, sound, animations and video delivered by some form of computer technology. A multimedia system is characterized by computer-controlled, integrated production, manipulation, presentation, storage and communication of independent information, which is encoded at least through a continuous (time-dependent) and a discrete (time-independent) medium."

The multimedia components that can be discussed included **text, images, voice, video, animations and data storage**.

Text

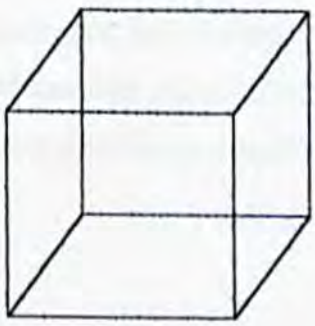
Below are some of the guidelines to be followed when using text in multimedia:

1. Make sure every word on the display screen is just the comfortable size to be viewed at. This does not mean we must try to enlarge the font size to as large as possible, but make sure it is comfortable to users' eyes. We can place different font sizes in a page to highlight certain important words, but there is a limit of maximum 3 font sizes to be used, anymore than that will make the page look untidy and messy.
2. Do not use a lot of different font typefaces. There is also a limit of maximum 3 font typefaces to be used in a page. Too many font typefaces in a page is very tiring to be read by the user.

3. The spacing between words, lines and paragraphs have to be adjusted accordingly if necessary. The arrangement of text directly affects users' reading speed their tendency to continue reading.
4. We can try different colours to make text more legible or stand out. Again we should not use more than 3 colours (including black and white).
5. For very important words like heading, title or key instruction, we can try other effects, such as drop shadows, making the text curved or blink to attract users' attention.
6. Finally, never try to highlight a line of sentences by using all capital letters because sentences in mixed case are easier to read compare to those are in just capital letters. For instance:
    - SENTENCES IN MIXED CASE ARE EASIER TO READ.
    - Sentences in Mixed Case are Easier to Read.

Images

Vector Graphics – Image vector are built up from basic drawing instructions such as line, triangle, rectangle, ellipse and so on. Many drawing packages can help the users to gather those basic drawing instructions to form image objects. For example, a 3-D transparent cubic box (image object) is made up by 12 equal length straight lines (basic drawing instructions).



Besides that, images also can be scanned in or digitized from photograph, piece of artwork or a video.

Bitmaps and Pixmaps – A frame buffer to store vector graphics. The images are stored in the form of pixels (reflects a matrix of dots over the computer display screen, a kind of digital data). The size of the frame buffer is depend on the system, system with high resolution will take up more space. For example, system with the resolution 1024 x 1024 and 24bit/pix will need a 3MB frame buffer. In the black-and-white system, frame buffer is called bitmap, but it is pixmap in the colour system.

Voice

Voice (audio) information can be represented digitally. To get a good representation of sound in digital format (digital sound), we need to sample its amplitude at a rate (samples per second, smp/sec) equal to at least twice the maximum frequency (Hertz, Hz) of the analog signal. For high quality sound like what recorded in the CD-ROM, 44100 smp/sec is the rate used. After sampling, the signal amplitude must be put in digital form. 16 bits per sample are usually used for each channel for stereophonic compact disk. In this case, more than 65536 levels of amplitude can be distinguished. Thus, without compression, digital voice requires 705600 bits of storage space per second for each channel. A CD is usually rated at the capacity of about 650MB, this lead to an audio capacity of about one hour of stereo sound per one compact disk.

MIDI (Musical Instrument Digital Interface): MIDI is a communication standard developed for electronic musical instruments and computers. It provides a very efficient way of recording the performance information required to play music. It is not digitized sound, but a series of commands which a MIDI playback device interprets to reproduce the sound.

Video

Digital Video – Digital video is stored in files on a hard disk or CD-ROM. It can serve over the computer network. Digital video allows the user to select randomly the specific clips. Using digital module takes slightly more time, but no more than identifying

photographs to be made into slides has taken. Recent improvements in hardware and software have allowed the quality of digital video to meet or surpass that of videotape.

Live Video Feeds – Provide interesting real-time object of multimedia links. Any television channel or live camera feed can be the object of a link. For example, the medical student can view an operation such as heart transplant live through their PC. This will show them the situation in the operation theatre.

Animations

Animations are the rapid display of images such that the pattern of image display causes the appearance of movement for the objects contained in the image. To create an animation, you need to produce the sequence of objects that are to be displayed and then display that sequence at a particular display rate.

Many animations display their image sequences in a looping fashion. The major parameter of an animation is the number of image frames it displays per-second. The more frames per-second that are displayed, the smoother the animation appears to be.

If an animation is not designed correctly, this often results in partial images being displayed. GIF images can be stored in interlaced and non-interlaced formats. The pixels of an interlaced image are stored in rows of pixels, but the rows are out of order. The pixels of a non-interlaced image are stored in the same order that the pixels appear on screen. One common problem with animation is that the animation flickers as each image is displayed. Double Buferring is a technique used to achieve smooth animations in Canvases or Panels. While the program renders one image on the screen, it can build the next image in an off-screen buffer. It is memory and CPU consuming but is unavoidable as soon as you want to animate something.

Data Storage

A computer makes no different with a typewriter if it does not provide data storage facility. From the quotation above reflects that how important data storage facility is to a

computer system. Computers mainly consist of two types of memory. The first one is primary memory (fast retrieval but more expensive, limited capacity) and next is secondary memory (cheaper, huge capacity but slow retrieval).

Below are two types of disk that serve secondary memory storage to store huge capacity of multimedia presentation products.

1. Hard Disks – Build in the computer's central processing unit (CPU), it was once very expensive but the price has dropped tremendously and the storage capacity has increased over the years. Nowadays, we can easily get a PC with hard disk's total storage capacity more than 5GB. Multimedia requires large amounts of disk space because a multimedia work usually consists of animated graphics and photos, audio and video that take up a lot of space. For a computer game like EMG For Algebra, the hard disk at least 20MB in a standalone computer is required.
2. CD-ROM – Compact Disk Read Only Memory were develop to store a range of data in digital format. The laser beam technology is used to write data and read the information stored in the CD-ROM through CD drive. A piece of compact disk can accommodate up to 650MB. Therefore, mostly computer games in the market can fixed in one CD-ROM.

2.4 Programming Languages

At its most basic level, programming a computer simply means telling it what to do, and this vapid-sounding definition is not even a joke. There are no other truly fundamental aspects of computer programming; everything else we talk about will simply be the details of a particular, usually artificial, mechanism for telling a computer what to do. Sometimes these mechanisms are chosen because they have been found to be convenient for programmers (people) to use; other times they have been chosen because they are easy for the computer to understand. The first hard thing about programming is to learn, become comfortable with, and accept these artificial mechanisms, whether they make ``sense" to you or not.

In fact, you should not worry if some (or even many) of the mechanisms used for programming a computer do not make sense. Many computer programming mechanisms are quite arbitrary, and were chosen not because of any theoretical motivation but simply because we needed an unambiguous way to say something to a computer.

In this chapter, I am going to review on some of the programming languages, which are popular and widely used today. The three very powerful programming languages that I am going to study after this are actually the Visual BASIC, Java and C.

Visual BASIC 6.0

Visual BASIC is a programming language that encapsulates the complexities of the windows application program interface (API) into easily manipulates object. Visual BASIC is a visual oriented programming language. It is one of the most flexible visual object oriented programming language. Visual BASIC has one to many relationships in which one instruction leads to a series of machine level instruction. It is a forth generation programming language with more English-like word such as If, Else, Dim and so on. Thus, it is relatively easier to learn, to use and to understand compare to those third generation languages.

Visual BASIC is an event oriented programming language. Event oriented language implies that an application waits for an event to occur before taking any action. Event such as presses a key on the keyboard or clicks a mouse button. With this event, the complete wait for a key press or a mouse click.

Beside, Visual BASIC is able to handle Structured Query Language (SQL) statement to access and manipulate database in Microsoft Access.

Visual BASIC has many versions, here I am going to focus on the sixth version. As Microsoft releases the sixth version of its award winning development tool, more than 50% of all professional developers are using the Visual BASIC language. The Microsoft Visual BASIC development system version 6.0 is the most productive tool for creating high performance components and applications. Visual BASIC 6.0 offers developers the ability to create robust applications that reside on the client or server, or operate in a

distributed n-tier environment. Visual BASIC 6.0 is the Rapid Application Development (RAD) tool available either as a stand-alone product or as a part of the Visual Studio 6.0 suite of tools.

Visual BASIC 6.0 Benefits:

1. Use the Visual BASIC 6.0 integrated Visual Database Tools and New Data Environment Designer to visually design Oracle & Microsoft Structured Query Language (SQL) Server databases and create reusable data access queries. Look at that - all without leaving the Visual BASIC environment.
2. Build server-side Web applications that are easily accessible from any browser on any platform with Visual BASIC 6.0 Web Classes. Program highly interactive web pages as easily as a Visual BASIC form with the new Dynamic Hypertext Markup Language (HTML) Page Designer.
3. Quickly develop rich data forms, or use the new integrated Report Writer to develop sophisticated, hierarchical reports, all with drag-and-drop ease.

JAVA

The JAVA programming language is designed to meet the challenges of application development in the context of heterogeneous, network-wide distributed environments. Paramount among these challenges is secure delivery of applications that consume the minimum of system resources, can run on any hardware and software platform, and can be extended dynamically.

The Java programming language originated as part of a research project to develop advanced software for a wide variety of network devices and embedded systems. The goal was to develop a small, reliable, portable, distributed, real-time operating platform. Here we are going to look at some of JAVA's characteristics.

Java does not have pointers - Slight hype and not a positive one at that. Java has nothing but pointers and native types. You need to allocate memory using the "new" operator but

you don't have to free it (automatic garbage collection will do that) and range checking is applied to arrays. You cannot refer to a variable's memory region (no pointer arithmetic) and you cannot have two types on the same memory region (union). For example:

```
MyClass A,B;          // I define my own data type and 2 instances of that type.
A = new MyClass();    // I must allocate space for A before using it.
B = A;                // B and A are now pointing at the same thing any change you
                      // will make to A will apply to B and vice versa.
int C,D;              // Two numeric variables
C = 1;                // C equals 1.
D = C;                // D equals 1;
C = 2;                // C equals 2. D still equals 1. This is because int is a
                      // native type to Java which means the default action is
                      // to copy it and not to create a new reference to it.
                      // if you wish to copy one of your types in a similar manner you can use:
B = A.clone();        // This will create 2 separate instances of the same type.
```

Java is easy for C++ programmers - Java is very easy for C++ programmers, but you need to get used to the differences that look identical in Java and C++ but have different meaning. The example above is the most obvious case, but there's more.

Java is object oriented (object based) - Java supports interfaces, which are a very pure object oriented tool. Java evangelists say java is above all regular object oriented languages and is in fact based on objects. Well, inventing new terms is a real cool way of adding features, and Java does add many paradigms, but it is debatable whether they are a part of the object paradigm.

Java is small - If the mean a small class file size, Java is very small and with JDK 1.1 Jar file format (a compression for multiple class files) it will be even smaller. Java was

optimized for size since it's supposed to run with low resources and low bandwidth. Java's APIs are definitely not small, and they are growing at a most unbelievable rate.

Java is secure - The bugs that were found in the early JDK were things that would have taken forever to be found if Java were not openly developed. Java deals with most security issues with the virtual machine concept, Java is secure but the issues involved are very complex.

Java programs can't access the disk - Java applets in today's browsers (such as NS 3) cannot access the local disk in any way (read or write). Navigator 4 will allow signed applets to be given permission by the user to access the disk. However, in future versions you will be able to throw restrictions at applets so as they will only see certain parts of your disk.

Java does GUIs - True Java is very good at GUI development, but it's not all that it does. Java has excellent command line interfaces, and developing command line Java utilities has a (possibly) bright future.

C

C is a relatively small programming language, but one which wears well. In fact, it is one of the most powerful programming languages today.

C's small, unambitious feature set is a real advantage: there is less to learn, there isn't excess baggage in the way when you don't need it. But, it can also be a disadvantage to certain people: since it does not do everything for you, there is a lot you have to do it yourself. However, to another group of programmers, this 'disadvantage' is viewed as an additional advantage: anything the language does not do for you, it does not dictate to you either, so you are free to do whatever you want.

C programming language is sometimes referred to as a ``high-level assembly language.'' This is because if a person has programmed in assembly language, he/she probably finds

C very natural and comfortable, if that person has never programmed in assembly language, he/she may be frustrated by C's lack of certain higher-level features.

The C language was designed this way to ensure that seemingly-simple constructions expressed in C would not expand to arbitrarily expensive (in time or space) machine language constructions when compiled. If you write a C program simply and succinctly, it is likely to result in a succinct, efficient machine language executable. If you find that the executable program resulting from a C program is not efficient, it is less likely due to something the compiler did behind which you have no control over.

As mentioned above, C imposes relatively few built-in ways of doing things on the programmer. Some common tasks such as manipulating strings, allocating memory and doing input/output (I/O) are performed by calling on library functions. Other tasks which you might want to do, such as creating or listing directories, or interacting with a mouse, or displaying windows or other user-interface elements, or doing color graphics, are not defined by the C language at all. But you can do these things from a C program by calling on services which are peculiar to your programming environment (compiler, processor, and operating system) and which are not defined by the C standard.

C's Pros:

1. C is small, with unambitious feature set.
2. Less to memorize, users are free to code whatever they want with their own creativity to combine and manipulate those limited features.
3. Ensure the degree of complexity of a program, avoid the seemingly-simple code from expands to huge and complex execution.
4. C seldom has mistakes behind the compiler, but it is likely the fault of the programmer.

C's Cons:

1. C does not try hard to protect a programmer from mistakes.
2. C has very limited built-in classes, the users have to 'do a lot a work'.

3. It is sometimes quite frightening to a new programmer because he/she may take a few hours to come out with a simple program, which is functioning well.

2.5 Database Authoring Tool

Database authoring tool means a software has been specially customized with the aim of helping developers to produce database. Mostly authoring tools required only some simple coding work by the user. Authoring tools are generally menu driven and the code is created automatically. The user might just do some modification on the codes if necessary.

Microsoft Access 97

Microsoft Access 97 is a powerful database management system (DBMS) that can be used to create relational database for storing information. Access tool has allowed non-programmers to work with database. Microsoft Access 97 enables users to create tables, edit data, and use queries to find the needed data with very little effort. Access includes wizards can do the work of designing data entry forms, reports, and mailing labels for users. Microsoft Access 97 has a module called Query. Structured Query Language (SQL) can be used in this module to retrieve, insert, delete or edit information stored in an Access database.

Microsoft Access 97 provides database power to give the information to the user to make better decisions. It integrates data from spreadsheets and other databases, and is the easy way to find answers, share information over Intranets and the Internet, and build faster business solutions. Microsoft Access 97 allows the user to generate analyze and create reports within hours. It integrates ease of use from the data entry point to printing in HTML.

For additional information, Visual C++ or Visual BASIC can access the Microsoft Access 97 database easily by referring to the path that indicates the location of this Access database file.

2.6 Other Application Software

Other than the coding and database design authoring tools, there are some other authoring tools needed in the process to develop EMG For Algebra. Since EMG For Algebra accommodates multimedia concept. Some multimedia authoring tools are needed to assist the process of designing various graphics, text, sound and animations to be implemented in the games.

Macromedia Director 8

Macromedia Director 8 Shockwave Studio is the world's foremost authoring tool for creating interactive multimedia. Developers rely on Director to create attention-grabbing business presentations, advertising kiosks, interactive entertainment and educational products. Director supports presentation of audio, graphic, text, animation and video. It allows the developer to import images, sound, movie and animation from external source. Director 8 by Macromedia is fine customized to assist even a non-programmer to come out with multimedia presentation. Macromedia Director 8's characteristics and functions are represented by icons make the job simple for those users who don't like to memorize commands and do coding.

Macromedia Flash 5

It is seems Macromedia Flash 5 is good at performing movies. Flash movies are graphics and animation for Web sites. They consist primarily of vector graphics, but they can also contain imported bitmap graphics and sounds. Flash movies can incorporate interactivity to permit input from viewers, and you can create nonlinear movies that can interact with other Web applications. Web designers use Flash to create navigation controls, animated logos, long-form animations with synchronized sound, and even complete, sensory-rich Web sites. Flash movies are compact, vector graphics, so they download rapidly and scale to the viewer's screen size.

Nowadays, we can probably watched and interacted with Flash movies on many Web sites, including Disney, The Simpsons, and Coca-Cola. Millions of Web users have

received the Flash Player with their computers, browsers, or system software; others have downloaded it from the Macromedia Web site. The Flash Player resides on the local computer, where it plays back movies in browsers or as stand-alone applications. The Flash movie has up to the extend - viewing a Flash movie on the Flash Player is similar to viewing a videotape on a VCR - the Flash Player is the device used to display the movies you create in the Flash authoring application.

Adobe Photoshop 6.0

Adobe Photoshop 6.0 is a powerful software capable of creating complex, attractive and beautiful graphic. It is easy to use and learn. Graphical image can be edited, cut copied, changed color mode and adding text. Photoshop is suitable to be used by professionals who recognize the importance of creativity. Adobe Photoshop 5.0 allows image like clip-art, scanned photo and image captured from video device to have its colour modified and enhanced. Besides, it is very effective to create small picture or banner, using Adobe Photoshop 5.0 is very simple comparatively.

2.7 Reviews On Other Computer Games

Review on other available shareware that applying multimedia technology gives me ideas on how EMG looks like, before I can go on with my own design of user interface and multimedia implementation of my project.

Educational Mathematics Games

I am going to list down some of EMG shareware reviewed. Those are among the good EMGs that we can get from the Internet or in the market, which can enlighten me to design good applications and user interfaces for my project. However, each shareware might have their strengths and also weaknesses, but I'll try to develop a better one, the shareware listed below are just to serve the purpose of reviews and references.

Geometry Blaster

This game claims to cover an entire year of high school geometry syllabus.

The story line and players' mission:

The Geometrons are trying to reduce the land of Dimensia from 3 dimensions to 2. Andi (an earth girl) and Zoid (a native of Dimensia) need your help to collect the five missing pieces of the Dimension Machine and save the planet.

Each of the five pieces is located in a different area. The Capital Building contains a concentration game where you match figures and their names. You climb up the N-Gon Mountains by correctly answering multiple-choice questions. You practice estimating angles in the Pit of Dispair as you bounce a beam off the walls to hit a target. In the Building of Truth, shapes are sorted according to whether they fulfill a given statement (such as "the interior angles add up to 360 degrees") or not. In the Proof Palace you will find geometric proofs in which the statements are out of order and you need to match them to their reasons. Once you reach the Dimension Machine you need to visualize how pieces fold up into a cube in order to put them in their proper slot.

The background information for the puzzles and other topics are covered in an online Geometry Handbook which offers 52 lessons as well as practice problems. Other online references include a glossary, a list of theorems, geometric constructions, a calculator, a geoboard, and tangrams.

AlgeBlaster 3

This game opens with an animated sequence that introduces the story:

A group of aliens from the planet of Quadratica are suddenly attacked by the Red Nasties. You help the Quadraticans by solving algebra problems in the four areas of the game.

In the Challenger you learn and practice algebra via video chalkboard, example window, practice window, or word problems. In the Decoder you match equations with written descriptions of equations. Correct answers provide pieces of a puzzle. When all the pieces are collected, you must decode them in order to select a correct vocabulary word. The Simulator provides practice with finding and naming points and slopes on a

coordinate grid with four quadrants. In Red Alert, you defend the buildings of a city by using a laser beam to blow up incoming space ships which otherwise explode upon landing.

Other Computer Games

The following is a list of other computer games shareware reviewed:

Minesweeper

Players' mission:

To find all the mines as quickly as possible without uncovering any of them.

How to play:

On the **Game** menu, click **New**. To start the timer, click any square on the playing field.

Notes:

The game area consists of the playing field, a mine counter, and a timer.

You can uncover a square by clicking it. If you uncover a mine, you lose the game.

If a number appears on a square, it indicates how many mines are in the eight squares that surround the numbered one.

To mark a square you suspect contains a mine, right-click it.

Strategies and tips:

If you are uncertain about a square, right-click it twice to mark it with a question mark (?). Later, you can either mark the square as a mine or uncover it by right-clicking again once or twice.

When you have marked all mines around a numbered square, you can quickly uncover all empty squares around it by clicking that square with both mouse buttons. If not all mines touching the square are marked, the uncovered touching squares will flash.

Look for common patterns in numbers, which often indicate a corresponding pattern of mines. For example, the pattern 2-3-2 at the edge of a group of uncovered squares indicates a row of three mines next to the three numbers.

Single Player Wheel of fortune Online

Like the TV show, the object of Single Player Wheel of Fortune Online is to correctly solve puzzles with category titles like "Fictional Character" or "Before & After". But unlike the TV show, you're playing this game solo.

A complete game is comprised of three rounds (one puzzle each) followed by a Bonus Round. A round concludes when a player either solves the puzzle or runs out of turns. There are five (5) turns allowed in each of the first three (3) rounds. The Bonus Round concludes when a player either solves the puzzle or the 60-second timer runs out. Players receive a cumulative score at the conclusion of the Bonus Round which can be ranked against the scores of other Station members.

BEGINNING TO PLAY

To start the game, spin the Wheel by clicking on the "SPIN!" button, or, if you're feeling lucky, try solving the puzzle. Spinning the Wheel determines the dollar value of your guess -- assuming you don't land on a "BANKRUPT" or "LOSE A TURN" wedge. Choose a consonant by clicking on a letter in the display below the puzzle board. If you guess correctly, the board will reveal your letter's position in the puzzle, and you'll receive the dollar value for every time that letter appears in the puzzle. If your guess was incorrect, a turn will be subtracted from the "Turns Left" counter. You do not lose money by guessing consonants incorrectly.

BUYING A VOWEL

Once you've accumulated enough money in your "Puzzle Bank", you can buy a vowel for $250. In order to buy a vowel, click on the appropriate vowel in the "Buy a Vowel" letter display. Buying a vowel always costs $250, whether that vowel appears in the puzzle multiple times or not at all. Note: You may select more than one vowel in succession as long as you have at least $250 in your "Puzzle Bank".

LOSE A TURN

There are four (4) ways to lose a turn: 1) choosing a letter that's not in the puzzle, 2) landing on the "LOSE A TURN" Wheel wedge, 3) landing on the "BANKRUPT" Wheel wedge, or 4) guessing an incorrect puzzle solution. If you lose five turns, then you lose any amount in your "Puzzle Bank" and the round is over. TIP: To check the status of your turn count, look at the "Turns Left" counter.

GOING BANKRUPT

In addition to losing a turn when you land on the "BANKRUPT" Wheel wedge, you lose all the money in your "Puzzle Bank" for the round only. You can rebuild your "Puzzle Bank" by making subsequent correct guesses.

FREE SPIN

If the Wheel comes to rest on the "FREE SPIN" wedge, you can win a "Free Spin" token by correctly guessing a consonant. If you're right, the "Free Spin" token appears beside the "SPIN!" button. The "Free Spin" token can only be used immediately after you lose a turn -- even if it's the last turn in the round. Using the "Free Spin" increases the number in your "Turns Left" counter, but does not add any value to your "Puzzle Bank". TIP: the "Free Spin" token carries over from round to round, but may not be used in the Bonus Round.

SOLVING PUZZLE

When you think you can solve the puzzle, click inside the "Solve the Puzzle" text window so your typing cursor appears, type in your answer, then click the "ENTER" button. If you're right, you'll gain everything in your "Puzzle Bank". But remember -- spelling counts! If the puzzle is "ABE LINCOLN" and you enter "ABE LINKEN", you'll lose a turn for an incorrect guess! Capitalization does not matter when solving the puzzle. A puzzle can also be solved simply by guessing all of its letters. TIP: The "Category" window may provide you a hint to solving the puzzle.

PLAYING ROUNDS

There are three standard rounds (puzzles) in a Single Player Wheel of Fortune Online game, followed by a special Bonus Round. In order to progress to the Bonus Round, the player must have a positive amount in the "Game Total." The current round is indicated by the "Round" counter beneath the puzzle board.

PLAYING THE BONUS ROUND

If you complete the first three puzzle boards with a positive score, you'll be rewarded with a "BONUS PUZZLE". In the Bonus Round, players are presented with a category and a puzzle in which the following letters are already revealed: R, S, T, L, N, and E. The player then makes a wager in the "Enter Your Wager" window -- anywhere from zero to the full amount in your "Game Total". The wagered amount will appear in the window above the "Bonus Round" sign. Once a wager is accepted, the player picks three (3) additional consonants and one (1) vowel from the look-up lists and clicks the "ENTER" button. This triggers two things: 1) it reveals the position of the selected letters if they are in the puzzle, and 2) it initiates a 60-second timer. The player has 60 seconds to guess the answer to the puzzle -- a red Time Left circle indicates the countdown. You may type as many guesses as you like in the allotted time; each guess must be followed by clicking the "ENTER" button. A correct answer rewards you with the entire wager value while you can have multiple incorrect guesses. If you do not correctly solve the puzzle before the timer runs out, your wager will be subtracted from the "Game Total".

SCORING AND WINNING THE GAME

The "Puzzle Bank" keeps track of your score for the current puzzle. Once you complete a round, the value of your "Puzzle Bank" is added to your "Game Total"; there is no way to lose money from your "Game Total" while playing the first three rounds. During the Bonus Round, your wager comes from your "Game Total" (after your response, the wager is either added to or subtracted from your "Game Total"). After the Bonus Round,

your Game Total becomes your score (not in real dollars, sorry). If it is high enough, your score might appear on one of the high scorer lists!

Comment On The Reviewed Computer Games

Thus all together 4 games I had reviewed. The first 2 games are same type of games with the proposed EMG For Algebra. The Geometry Blaster trains the players' skill in applying their knowledge on the topic of geometry, while AlgeBlaster 3 emphasizes on the algebra.

The last 2 games are basically to serve the entertaining purpose, to let players to have to have fun on it and relax themselves.

Though those games are popular and have lots of admirers, it still possessing both strengths and weaknesses, which allow me to pass some comments to it.

Strengths:

- The story line of AlgeBlaster 3 and Geometry Blaster are very attractive especially to the kids (boys) and teenagers.
- Variety of functions and modules in the Single Player Wheel of fortune Online, players can use their creativity to work out their strategy to probably solve the puzzle and explore to more advance modules.
- Minesweeper is a relatively small game and mostly home PC has installed this game, therefore this game has became a "snack" game those people who has to face the screen for long hours a day.

Weaknesses:

- The story line of AlgeBlaster 3 and Geometry Blaster tends to be too male-oriented, usually girls won't like those jargons like 'shooting', 'planet' and so on.
- Those games are too 'mouse-clicks oriented', the players are just required to use the mouse-button to due with the game. This might indirectly cause the players to neglect the usage of keyboard and over depend on the mouse.

- The graphic resolution for those online games are to low makes the screen displays become blurred when players trying to enlarge the games' screen.
- Lack of different level of difficulty: Single Player Wheel of fortune Online does not provide levels of difficulty of the puzzle, all standards of players are required to play from the beginning with the randomly pick puzzles.

2.8 Conclusion

After a survey has been carried out on various authoring tools available, the previous work done by other researchers and other computer games, it gives me some ideas about how an educational computer game is look like and the outline of EMG For Algebra. No matter what tools used and what runtime environment selected, the system has to have the ability to support multimedia application.

Visual Basic 6.0 or other versions might be a suitable authoring tool to help in the system prototyping and coding process, since Microsoft Access 97 database development tool works well with Visual Basic, it can be used to create system's database. Macromedia Director or Flash can be used to create multimedia effects in the games.

After reviews on other similar package, there are strengths I must try to inherit, and also weaknesses that I must avoid. This is to ensure my proposed system, which is the EMG For Algebra will be a computer game that can satisfy users' need.

CHAPTER 3: METHODOLOGY

3.1 Introduction

Careful planning is required for project development process and the process of determining the project life cycle. This will help to avoid the problem such as schedule slippage, cost overrun, poor quality of product and high maintenance cost after the system implementation. Thus, the most important activity in the project planning phases is actually determining or planning the project development process.

3.2 Project Development Process

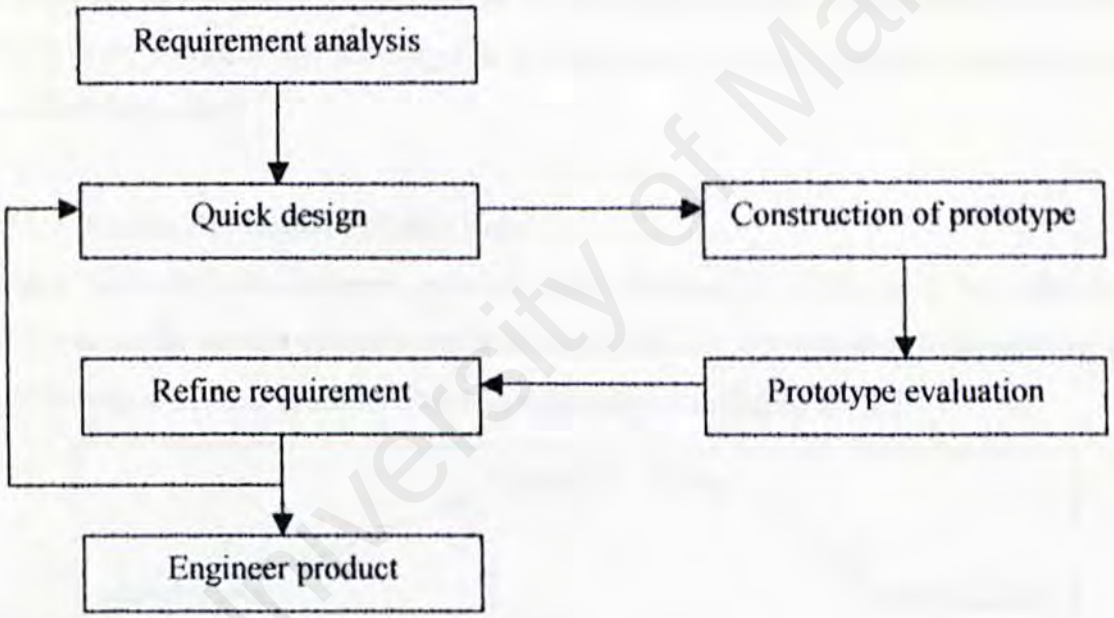


Figure 3.1: Software prototype model

Requirement analysis: To develop an abbreviated representation of the requirements.

Quick design: An abbreviate design specification is created for prototype. This leads to the construction of a prototype.

Construction of prototype:	Prototype software is developed, tested and refined.
Prototype evaluation:	The user evaluates the tested version of the tested prototype and user will suggest modification.
Refine requirement:	Justify the requirement or add in new requirements. The customer need is specified here.
Engineer product:	Activity of 'quick design' to activity 'refine requirement' are repeated until all requirements are formalized or until the prototype has evolved into a production system.

Project Developing Strategy

Project developing strategy is based on the software prototype model showing above. Prototyping development is an idea of developing an initial implementation, exposing this to user comment and refining this through many versions until an adequate system has been developed.

3.3 System Development Life Cycle

System (software) development generally takes the form of a life cycle. We refer to this life cycle as the system development life cycle (SDLC). All systems go through the same generic stages in their lifetime. The stages are shown in the figure 3.2.

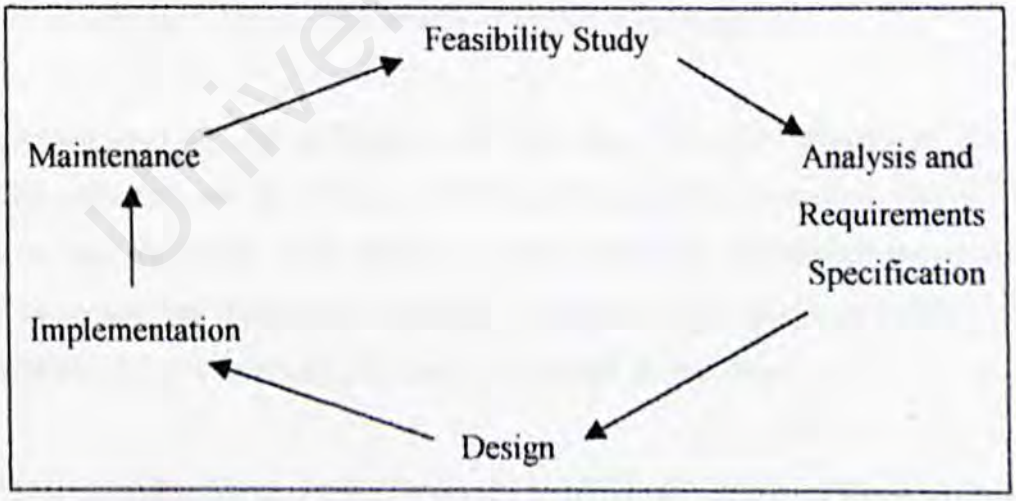


Figure 3.2: System development life cycle (SDLC) stages

The software engineering process consists of a set of steps that encompass methods, tools, and procedures. These steps are often referred to as software engineering paradigms or software life cycle models. A model chosen by the developers is based on the nature of the project and applications.

In this chapter, I will emphasize on Waterfall model, which to be used in my project development process.

Waterfall Model

The waterfall model was derived from engineering models to put some order in the development of large software products. It consists of different stages, which are processed in a linear fashion. Compared to other software development models it is more rigid and better manageable. The waterfall model is an important model, which is the basis of many other models.

Waterfall model is an engineering model designed to be applied to the development of software. The idea is the following: there are different stages to the development and the outputs of the first stage "flow" into the second stage and these outputs "flow" into the third stage and so on. However, this idea sometimes is not practical but it is common for mostly system development process.

There are usually five stages in this model of software development:

Requirements analysis and definition.. In this stage the requirements of the "to be developed software" are established. These are usually the services it will provide, its constraints and the goals of the software. Once these are established they have to be defined in such a way that they are usable in the next stage. This stage is often preluded by a feasibility study or a feasibility study is included in this stage.

System and software design. In this stage the established requirements, flowing from the first stage, are identified as software or hardware requirements. The software

requirements are then translated in such a way that they can be readily transformed into computer programs.

Implementation and unit testing. This is the stage where the computer programs are created. Each program is called a unit, and unit testing is the verification that every unit meets its specification.

System testing. All the units are combined and now the whole is tested. When the combined programs are successfully tested the software product is finished.

Operation and maintenance. Most software products include this stage of the development. It involves correcting errors that have gone undetected before, improvement and other forms of support. This stage is part of the life cycle of a software product, and not of the strict development, although improvements and fixes can still be considered as "development".

These steps are the main stages. There are also sub-stages, within each stage, but they differ from project to project. For example for EMG For Algebra system the requirements stage is divided in a feasibility study, a synthesis on other similar package, choosing authoring tools, a definition of functional and non functional requirements, and the software and hardware requirements specification stage.
It is also possible that certain software projects require the adding of an extra stage all together, or the splitting of one in two stages. However all the different waterfall models have the same underlying idea; the idea that one stage provides outputs which can be used as the input for the next stage. There thus is a linear flow amongst the stages. The progress of the software development, using the waterfall model, is thus easy to find out. A common way to look at the outputs of a certain stage and see whether or not they are finished in time, thus seeing how far the overall progress is.

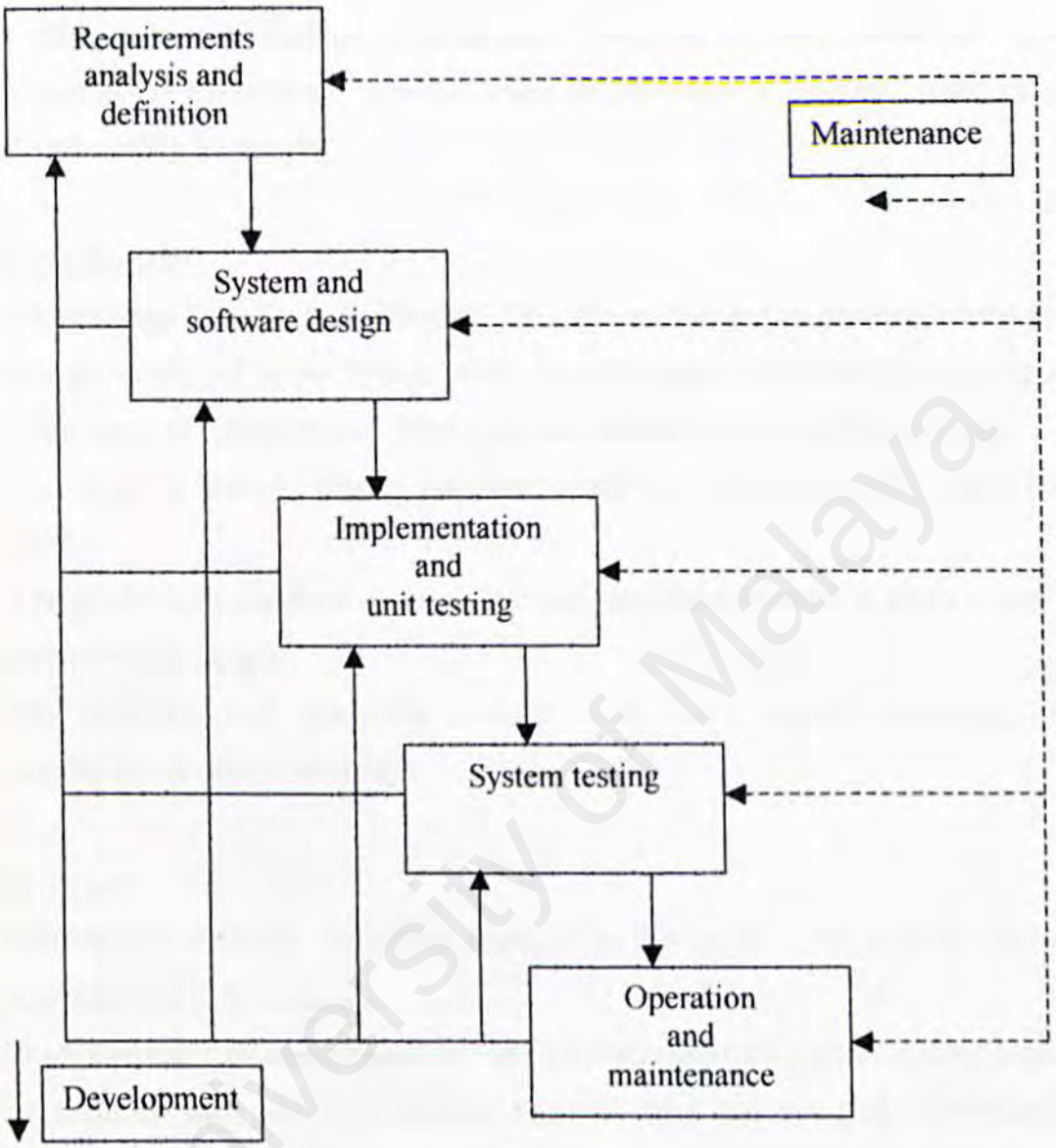


Figure 3.3: Waterfall model

It has to be noted that the software development process is not as linear as it seems. When errors in later stages are found, they are often fed back to a previous stage and the development is set back to that stage again. Since this is a managing nightmare, it often occurs that problems are ignored, left for later or programmed around. This feed back makes for a waterfall with information flowing both ways: down through the stages when

something is made, and up through the stages when something goes wrong, or feedback is given. Also many processes are frozen when it is not yet the time to deal with them. This has lead to the development of other, more flexible models later on. However, the Waterfall still widely used today.

Why choose Waterfall:

1. The package has circulation factors. This allows changes in previous phase even though we are not in the current phase. In other words, this model recognizes the importance of backtracking (feedback) and iteration in the software process.
2. It is easy to identify project milestones and easy to separate one stage from another.
3. The process development is systematic and sequential because it follows step by step (stage by stage).
4. The method is well wide used. In other words it is a popular methodology for system development nowadays.

Waterfall's *cons*:

1. Due to the difficulty in adding changes to the model, Waterfall is a slower development SDLC model.
2. It implies that any stage should be frozen before continuing with the later stage.
3. It assumes that user requirements can be precisely specified. Unfortunately customers rarely know precisely what they want, and software engineers rarely understand the business context (not for EMG For Algebra) of their customers.

3.4 Prototyping

Prototype: a small-scale, incomplete, but working sample of a desired system. Prototypes cater to the "I'll know what I want when I see it" way of thinking that is characteristic of many users and managers.

Prototyping is particularly valuable in the following situation:

- There is scope for user creativity to improve the system.
- Users are unsure of exactly what they want
- The system changes a basic business operation.
- An end-user dialog should be tried out with the users to see if it can be improved.
- The users do not understand all the impacts of the new system.
- The functions are subtle, and the users understand them better than the analysts.
- Screens and reports should be checked with management to see if they can be made more useful or easy to use.
- The users have difficulty expressing all the system requirements.
- The prototype may act as a catalyst to elicit alternative ideas.
- The relative merits of alternative solutions need to be explored.
- Experimentation may be done to achieve better business practices.

There are ways to implement the prototyping phase into the software (system) development process. Here I am going to list down some of the very common prototyping approaches (or models).

Rapid Prototyping

Rapid prototyping is a process that enables the developer to create a model of the software to be built. It is a working model that is functionally equivalent to a subset of the target software. The idea is, system developer first builds a rapid prototype and then lets the user(s) to interact and experience with it. If the user(s) satisfy with the prototype, the development process than can be continue or otherwise the developer has to modify the system design.

Throwaway Prototyping

The most significant difference between throwaway prototyping model with the other models is its prototyping process and production process are clearly separated.

A quick design based on the inputs, user interface, outputs and those aspects of the software that will be visible to the user, leads to the rapid construction of a prototype. The user than gives the feedback and the prototype is refine accordingly. This iteration process is turned on until the user satisfies with the system design. Once the prototype is finalized, it is thrown away, the actual production of the operational system begins.

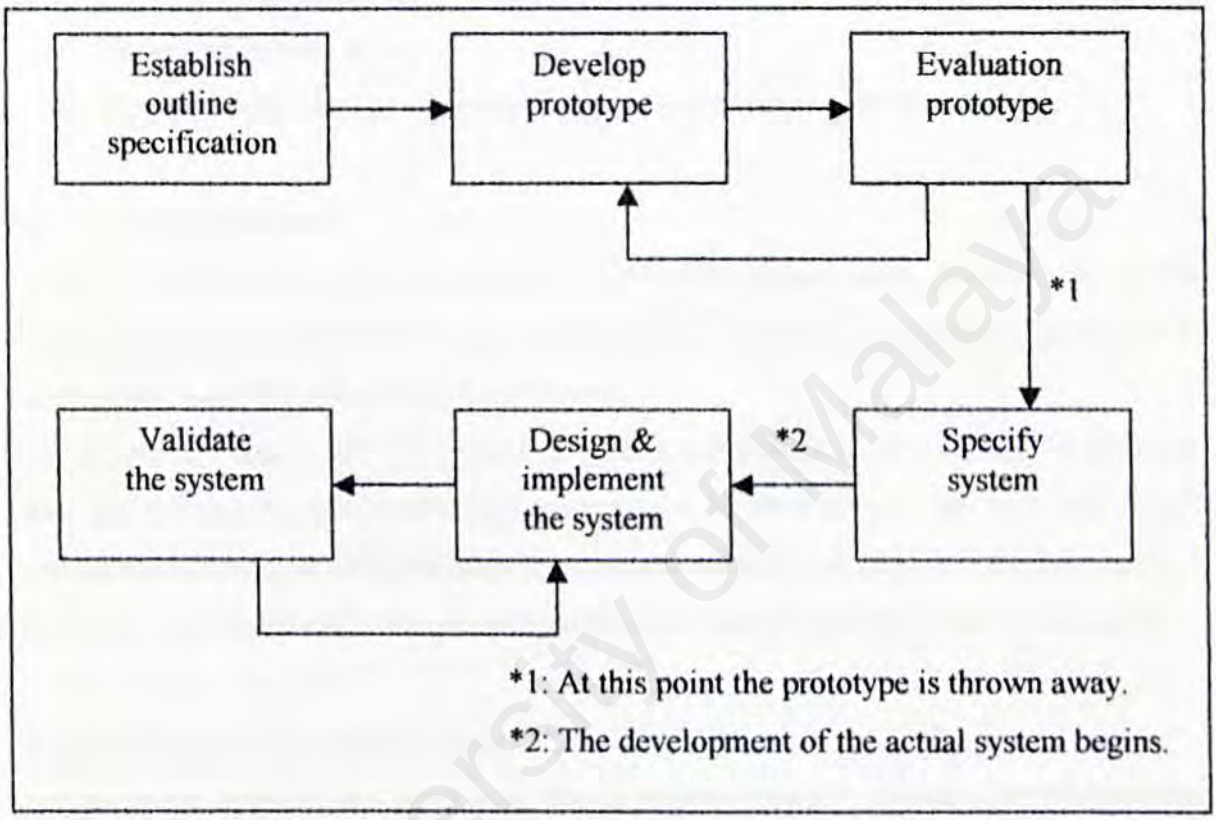


Figure 3.4: Throwaway prototype model

Evolutionary Prototyping

In this approach the prototyping and the production process are merged. This means, the prototype gradually evolves to become the final product. This model is useful in situations when it is extremely difficult (or impossible) to establish detailed user requirements such as in user interface design and Artificial Intelligent applications.

The benefits of prototyping:

- With a good tool, it is easier (faster) to build a prototype than to build paper specification.
- Users understand and react to prototype far better than to build paper specification.
- Enables errors and weaknesses to be detected before the expensive design and programming are done.
- It encourages users to contribute creative input into the design process.

3.5 Project Schedule

Project management is very important for the success of a project. A successful system project will be developed if developers understand the scope of the project, the task to be accomplished and the schedule to be followed.

The project schedule is the operating timetable of the project. It serves as the fundamental basis for monitoring and controlling project activity. In a project environment, proper scheduling functions is of paramount importance because projects lack the continuity of day to day operations and often present much more complex problems of coordination.

Project Schedule For Current Project

The proposed project (my final year thesis project: build a Educational Mathematics Games (EMG) For Algebra computer game system) will be carried out in two stages, which each stage has to be completed within the period of one semester respectively.

The first stage of the project is more on the requirements analysis and definition phase, and also the system and software design phase, according to the Waterfall SDLC model. Therefore, the tasks have to be done in this stage are to determine the scopes and objectives of the project, project planning, literature reviews, system analysis, system design and also start to build a system prototype.

In the second stage, the project will carry on with the coding, unit testing and implementation phase, the system testing phase, and the operation and maintenance phase

(as recommended in the Waterfall SDLC model). The tasks to be performed in the second stage are evaluates system prototype, system coding and testing, system implementation and maintenance.

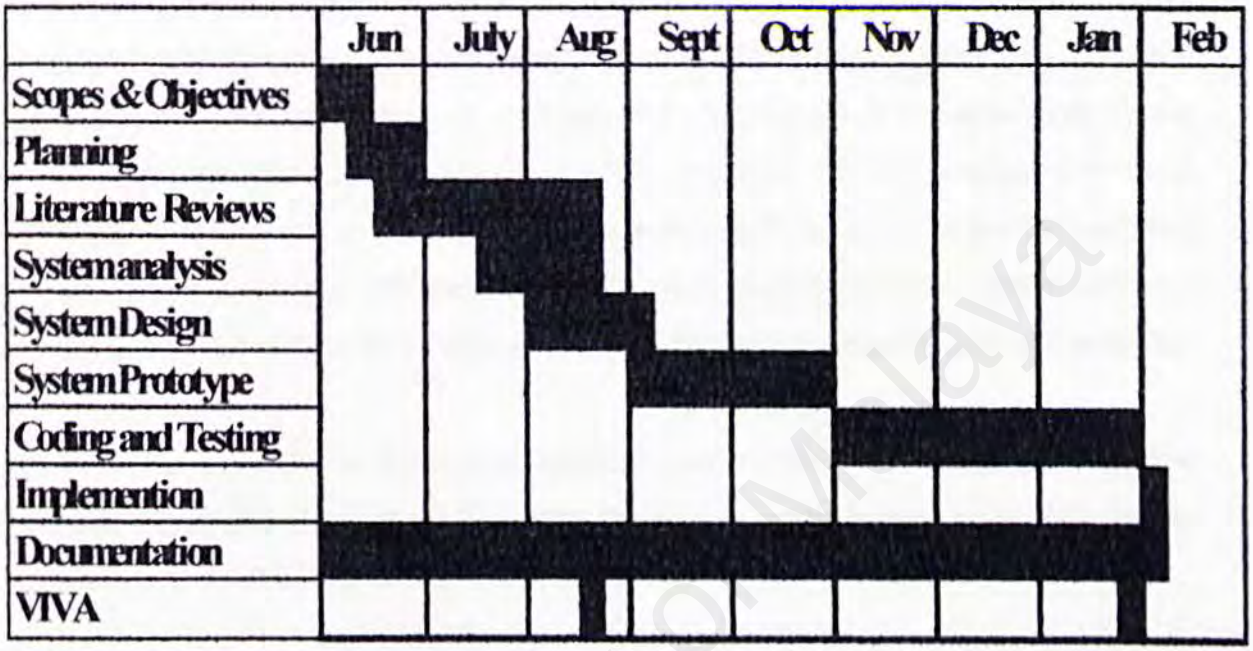


Figure 3.5: Project schedule

CHAPTER 4: SYSTEM ANALYSIS

4.1 Introduction

System analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those component parts work and interact to accomplish their purpose. The sentence above is one of the definitions for the term 'system analysis'. System analysis is very important for the developers to build effective, complete and consistency system models, and increase the productivity and product quality relatively. Different types of system analysis methods and models are recommended to tackle different type of problems and also different aspect of a problem.

Model-driven analysis: The drawing of pictorial system models to document and validate both existing/proposed systems. Ultimately, the system model becomes the blueprint for designing and constructing an improved system.

Structured analysis: A model-driven, process-centered technique used to either analyze an existing system, define business requirements for a new system, or both. The models (pictures) illustrate the system's component: processes and their associated inputs, outputs, and files.

Object-oriented analysis (OOA): A model-driven technique that integrates data and process concerns into constructs called objects. OOA models (pictures) illustrate the system's objects from various perspectives such as structure and behavior.

Accelerated analysis: Accelerated analysis approaches emphasize the construction of prototypes to more rapidly identify business and user requirements for a new system.

Rapid architecture analysis: An attempt to derive system models (as described earlier in this section) from existing systems or discovery prototypes.

Reverse engineering technology: Reads the program code for a database, application program, and/or user interface and automatically generates the equivalent system model.

Joint requirements planning (JRP): The most significant benefit of this technique is facilitated workshops to bring together all of the stakeholders to jointly perform systems analysis.

In common there are four systems analysis phases as listed below;

- Preliminary Investigation Phase
- Problem Analysis Phase
- Requirements Analysis Phase
- Decision Analysis Phase

4.2 Synthesis of Other Systems

I had reviewed on several computer games (including other educational mathematics games) as documented in Chapter two. Below are some of the details of the lacking part of those games which to be avoid in my proposed system.

Lack of multimedia element: the Minesweeper virtually does not apply the concept of multimedia, no sound, no graphic movements, nothing, just with some simple icons and some static graphics.

Limited modules: Some of the games do not provide various modules or various standards of game. Therefore, different standard of players with different preference are force to explore the same module or functions every time they play.

Less connection to the school syllabus: The entertaining computer games sometimes can be very attractive to the kids and teens, but usually their parents will strongly against them if they are addicted the those games. This is because the games have not connection to the school syllabus, the parents might think their children learn nothing but just wasting time and neglecting their homework.

Consisting unhealthy elements: some of the game consists gamble element like the 'Wheel of Fortune' because the game objective is not just to solve the puzzle, but win dollars by spinning the wheel! Next, some games with the story line like AlgeBlaster 3 and Geometry Blaster, which suppose to be educational games for kids and teens, but tend to be a bit aggressive and too male-oriented, usually girls won't like those jargons which appear a few times in games' introduction like 'shooting', 'save the planet', 'hit the target', 'attack by aliens', 'defend' and so on.

4.3 Analysis: EMG For Algebra

System analysis is carried out to study and identify users' requirements and needs toward the system. Besides, the feasibility study and technical analysis of the system are also very important in assisting to allocate the available resources for each functional and non-functional requirement. Thus, various system analysis techniques are applied in the process.

Internet surfing: Internet has become the largest information warehouse in the world. It consists virtuously any kinds of information that men can think of. I surfed Internet for the references on any ambiguities that arise during the entire development period. I downloaded games, the information of various approaches toward software development process and so on, to be the content of literature review.

Books and hardcopy reading materials: Books and other reading materials provide information needed to develop the system. This includes information on algebra,

programming language, authoring tools, multimedia applications and related mathematics problem solving skills.

Document room: Another source for gathering information is the document room (bilik dokumen) in Faculty of Computer Science and Information Technology. Here, some documentation done by the seniors in the past few years were found to be relevant to my project title.

Interviews: Few interviews have been carried out informally to find out some users' expectations towards my proposed system. I have talk to some of my course mates about my project and listen to their opinions towards EMG For Algebra. In the coming days, I am going to interview more people including my supervisor and moderator, and some of my product's target users.

Project development process

Waterfall model is chosen to be this project's system development life cycle because it is appropriate to this project, based on those Waterfall's characteristics that stated down in the previous chapter. The arguments that show Waterfall model is suitable for EMG For Algebra project development are:

1. The project is relatively short and easy to understand.
2. The scope of this project is well understood.
3. The project requirements are well estimated before the actual development process starts.
4. The developer (myself) well understand how does the system is going to look like.
5. The project risks have been accessed and are considered to be low.
6. The users do involved in the whole prototyping process.

4.4 Functional Requirement

Functional requirement is a description of activities and services a system must provide, it describes the interactive between system and the environment. Functional requirement are functions or subsystems that are mandatory to the system. The absence of the functional requirement will make the whole system incomplete. In the case of EMG For Algebra computer games system, the functional requirements are listed as below.

Load The System In Any Recommended Environment

EMG For Algebra must be able to be installed or loaded in any recommended environment such as in the Window98 or Window2000 operating system. Any players can enjoy the games, provided they have the recommended hardware system with proper embedded software (those needed during program runtime environment) in front of them.

Three Gaming Chapters

The game system must included the 3 gaming chapters as mention in the first chapter of this report, which are

Chapter one: + (add) and – (subtract) mathematical operation for algebra

Chapter two: * (multiply) and / (divide) mathematical operation for algebra

Chapter three: Polynomial

Help on EMG For Algebra Module

This module must be exists and ready to help the users by:

- Brief them about what EMG For Algebra is and the content of this game system.
- Tell the users about all the control buttons, shortcut keys used and its functions.
- Tell the users and briefly show them how to play the games and the objective of the games.
- Provides them some other guidelines and tips when playing this game.

Exit Game Module

To enable users quit from the EMG For Algebra whenever they wish to leave the system. The computer environment must be able to recover to the initial condition (before users open the game file) after users quit from the game.

Five Modules In Each Gaming Chapter

In each gaming chapter, which is chapter 1, chapter 2 and chapter 3, five modules including four gaming modules and one exit chapter module must be provided. Those gaming modules are multiple choice (objective) question, subjective question, falling bricks and puzzle module. 'Exit chapter' module enables users to quit from the chapter, which he/she is currently in and back to the main menu page.

Different Level of Questions' Difficulty

The questions in every gaming module of every chapter must vary in the level of difficulty. The standard of questions must range from suitable for form one students up to form 6 (college as well) students. Since we must cater for each level of our target users and allow them to improve their skill through playing the games.

Retrieve Questions and Answers From Database

The program must be able to retrieve questions and the correct answers respectively from the database. The system must be able to give respond whenever players submit their answer or the time limit is up, either certifying players' correct answer or showing the correct answer if the user failed to solve a question.

Players' Reward

When player(s) answer a question correctly within time limit if there is, marks should be added to the total marks column.

When player(s) answer series of questions correctly, bonus marks should be provided. When player(s) collect total marks reach certain preset amount, a key should be added to the player reward box.

Vice versa, part of the rewards should be removed (minus marks or reduce the total number of key) if they fail to answer the question correctly.

Usage of Multimedia

As stated down in the project objectives, the concept of multimedia must be implied into the EMG For Algebra. This will include attractive graphics, sound and animations.

4.5 Non-Functional Requirement

Non-functional requirement is essential definition of the system properties and constraints under which a system must operate. It is a description of other features, characteristics, and constraints that define a satisfactory system.

Mostly system users might expect certain degree of non-functional requirement. Some of the non-functional requirements for EMG For Algebra are user friendliness, efficiency, short loading and respond time, reliability, accurate, modularity, maintainability and expandability. For those online systems, the stakeholders might consider a few more non-functional requirements such as the accessibility and security.

User friendliness

For a system to be popularized, it must be easily understood by the users. The users need not to know what happen behind the system but through the system's user interface, users are supposed to get whatever they want easily.

Here are some 'scheme's provided to measure whether a system is user friendly or else:

- Consistent, in terms of screen design and error messages displayed.
- Accommodation of any level of user. In the case of EMG For Algebra, it must be user friendly to all target users, the students, teachers as well as the parents.
- Appropriate error handling with associated error messages.

- High degree of understandability and avoid too much of memorization of events and commands for the users.

Efficiency

A system is said to fulfill the efficiency requirement when its process or procedure can be called, accessed and functioning well to produce outcomes or output at a pace or speed acceptable by the users. Furthermore, all that has to happen in an unlimited of times after the system implementation whenever the users need it. The outcomes of the same process or procedure with the same input must be similar every time being called.

Short loading time and respond time

Normally, everyone likes the system to respond fast. Thus a system must be able to provide short loading time and respond time (more critical if the system is online). Slow loading and respond time might cause the users to wait and discourage them from using the system again. However, the system's performance sometimes depends on the hardware used.

Reliability and accurate

A system must be able to produce accurate results and can be trusted by the users. This is most critical for the military, medical, financial, banking and scientific research systems. However, it is also very important in a computer aided learning system and also the educational games system. Take the EMG For Algebra educational computer game system as an example: all the answers compatible to all the questions in the games must be always correct and accurate.

Modularity

Modularity means the system is broken into small modules so that distinct functions of objects could be isolated from one to another other. This will make the system testing and

maintenance process easier because the processes can be done portion by portion and not involving the whole system.

Maintainability

This may be defined qualitatively as the ease with which software can be understood, corrected, adapted and enhanced.

Expandability

Expandability is discussed based on the degree to which system architecture, data or procedure design can be extended and enhanced after the system is implemented.

4.6 Choosing Development Tools

One of the most important choices to be made during the process of software development is that of which software tool should be used. A correct choice might lead the development process to a more comfortable and lower risk position. This is because not every software development tool in the market are general purpose, some tools might very useful in developing certain software (system) but not the other one.

This section examines development tool selection criteria by take a comparative look at the characteristics, strengths and weaknesses of those tools, to decide the most suitable tool(s) to be used in the project development process.

System Coding and Programming

Visual BASIC	JAVA	C
4GL programming language	3GL programming language	3GL programming language
Object oriented, with a lot of built in functions	Object oriented, with a lot of built in classes and applets.	Limited built in functions

More suitable in developing software with many control functions and GUI	More suitable in developing online system	More powerful in developing large program and operating system
Recommended by supervisor to develop the EMG For Algebra	Recommended by supervisor to develop the EMG For Algebra	Not mentioned
Easy screen design	More difficult compare to Visual BASIC	Difficult

Table 4.1: Comparative look between Visual BASIC, JAVA and C

Hence, I choose Visual BASIC version 6.0 as the coding authoring tools of my proposed system due to its suitability.

Database System

Since Microsoft Access 2000 database management system development tool works well with Visual Basic authoring tool, it is chosen to create EMG system's database, to make the process of data retrieving fast and easy. Besides, Microsoft Access 2000 is very easy to learn and wide used today.

4.7 Hardware and Software Requirement Analysis

Any software and hardware that supporting a system must be compatible in order for the system to be perform well. The hardware and software listed below are the basic requirement to develop and run the EMG For Algebra.

Hardware

Facility	Usage
System (PC) using an 500 MHz processor or above	In order to support all the selected development tools and support the EMG For Algebra in the runtime environment
128MB of memory	Be able to load all selected authoring tools and to run the proposed system.
At least 20MB of available hard disk space	To store all those authoring tools and the stand-alone game system.
Keyboard and mouse	User input
Multi-media speaker	Sound effect

Table 4.2: Hardware requirements and the usage of the facilities

Software

Facility	Usage
Windows 98 or Windows 2000 operating system	Apply in both development and runtime environment.
Microsoft Visual Basic 6.0	To develop the hardcode and graphical user interface
Microsoft Access 2000	To create database
Mp3Trim	To edit sounds
Adobe Photoshop 6.0	To edit and create graphic images

Table 4.3: Software requirements and the usage of the facilities

CHAPTER 5: SYSTEM DESIGN

5.1 Modules of EMG For Algebra

EMG For Algebra consists of 3 chapters, each chapter consists of 4 gaming modules and 1 exit module. Above all chapters and modules, there is a main menu user interface designed for the users to select which chapter they prefer to explore. Similar to most windows base software, two additional buttons are added in the main menu, there are 'Help on EMG For Algebra' button and 'Exit Game' button.

Therefore, there are 5 buttons in the main menu user interface. Users are required to click on the button to activate the command embedded behind the button.

The 3 chapters and 2 additional commands are:

Chapter one:

This chapter emphasizes on the addition (+) and subtraction (-) mathematical operation unto algebra. Various forms and different standards of mathematical equation problems will be posted to the player(s).

For example, a question can be as easy as

$$4 + 6 = A, \text{ find } A.$$

or a question can be at the expert level such as

$$4r + 3s + 6p = 32$$
$$3r + 3s + 17p = -98$$
$$10r + s + p = 1,$$

find $p - s$.

Chapter two:

This chapter emphasizes on the mathematical operation unto algebra. Since this chapter is chapter two, I assume the player(s) had already gone through chapter one or at least know

how to solve addition and subtraction mathematical equation for algebra. Thus, the problems posted to the player(s) might include add, subtract, besides multiply and divide mathematical operations. Various standards of mathematical equation problems in different forms will be posted to the player(s).

For example, an easy question can be like this

$4f = 8$, find f.

or a question can be at the expert level such as

$4rs + 6p = 3$

$3rp - 17p = 6$

$r + sp = 1$,

if $p = 4$, find rps

Chapter three:

This chapter is going to emphasize on the polynomial equation for algebra. Polynomial means the index of algebra in the equation can be other than one (more than 1 or less). Various solution methods are recommended in the textbook to find out the actual value of algebras in a polynomial equation. Basically, the players' mission in this chapter is only to give final answer of the polynomial question, the system does not require player(s) to show the works of getting answers, all of them are encouraged to think out of their own even faster way (but logical and acceptable) of polynomial equations' solution method. Thus develops their research spirits and obtain better understanding for the topic.

Again, the players are assumed to master all the four basic mathematical operations unto algebra equation before they enter chapter three.

One of the question on polynomial equation:

Solve $3p^2 + 6p - 7 = 2$.

Help on EMG For Algebra:

The 'Help' module here will provide information on

- About EMG For Algebra

- The control buttons and its functions
- How to play EMG For Algebra
- Other guidelines and tips

The system user may click on one of the title to obtain respective information from the display screen. After reading the texts, the user may click on the 'OK' button to quit this module and return to main menu. This is to assists the new players, and also to serve parents and teachers referencing purpose while guiding their children or students.

Exit Game:

This button enable users to launch the terminate command and quit from the EMG For Algebra. When user clicks on this button, a prompt up box will appear for the user to reconfirm his/her action.

When the user enters one of the 3 chapters, each chapter then have 5 modules, which are objective (multiple choice) module, subjective module, falling bricks module, puzzle module and exit chapter module.

In order to introduce the content of every module and the way to play this game, I would like to take ***chapter two*** to assists my explanation.

First, a user (player as well) starts this EMG and clicks on the 'Chapter two' button in the main menu page, another page (chapter menu interface) appears and lists down 5 buttons on it, each represent one module as showing below.

1. Multiple Choice question

This module challenges the player on multiple choices questions. There are 10 questions for each level of difficulty, each question followed by 3 to 5 choices of answers, the player is required to pick the correct one within one try.

For instance:

Question: $4f = 8$, find f.

Answer: A. 0 B. 1 C. 2 D. 3 E. 4

For this case, the correct answer is C. Therefore, the player have to click on the 'C' button within the time limit in order to score.

2. Subjective question

This module challenges the player on the questions with blank answer box after the question. The player has to type in the correct answer as quickly as he/she can. Similar to the first module, there is time limit here. However, it is not for the individual question but overall limitation for one level of difficulty. With the time limitation, the player may try to solve as many questions as possible (maximum 8 question per level). Besides, user can proceed to the next question if he/she wishes to leave the previous question unsolved, by clicking the button labeled "skip to the next question".

For instance:

Question: if $\frac{4def8aa}{2a^2fde} = y, y = ?$ time left : 18 second(s)

So, player has to type in the answer, which is 16 into the answer box, in order to score, then press 'Enter' key or space bar. The cursor is by default placed inside the answer box.

3. Falling bricks

This module tests player's ability of solving more than 1 question simultaneously and working under highly pressure environment. Although no time limit here, there is always more than 1 bricks (each contains answer) falling from the upper part of the screen throughout the game. If the player not able to move the bricks quick enough and accurate for every single brick throughout the game, the brick will fall on the wall improperly. Then the player's total marks will be reduced.

Thus, the player has to have great confidence towards his/her own decision and skill.

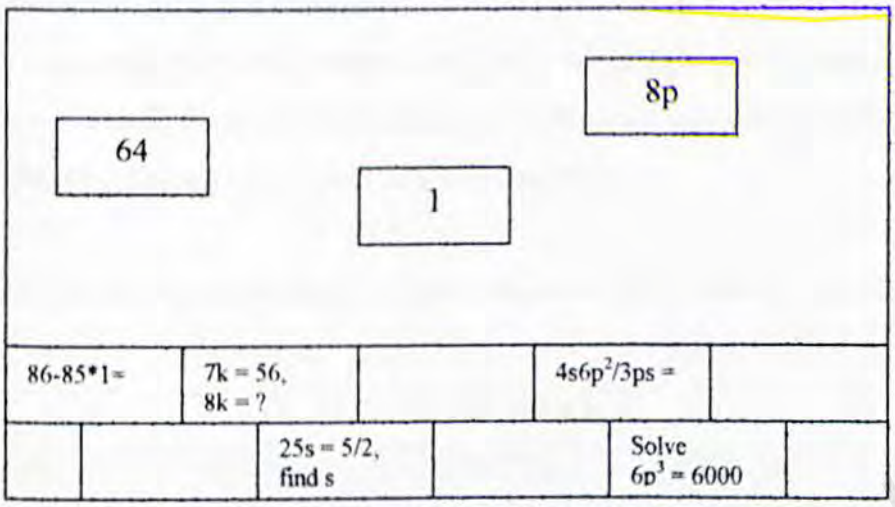


Figure 5.1: Sample of bricks and wall in the Falling Bricks module

4. Puzzle

In the puzzle module, each Puzzle pieces that contains number, algebra or mathematical operator are placed surrounding the puzzle board, which contains various forms of algebra equation, placed horizontally and vertically on the board, with a few blank boxes in between the equations. Player is required to place the puzzle pieces into the board to replace those blank spaces respectively. No time limit is implemented in this module.

5. Exit chapter

This module enable user to quit from the chapter which he/she entered. However, the user only back to the main menu instead of quit the system.

In order to suit different standards of players and to allow players keep on exploring the games, each module then has 3 levels of difficulty.

5.2 Pyramid Analysis Diagram

In order to develop the user interface for EMG For Algebra, all chapters and modules in the program should be listed down in a proper diagram according to their hierarchy. The type of diagram chosen is pyramid analysis diagram.

However, all relevant information is listed down in a table before move to the diagram;

MAIN MENU				
CHAPTER ONE	CHAPTER TWO	CHAPTER THREE	HELP ON EMG FOR ALGEBRA	EXIT GAME
Multiple choice	Multiple choice	Multiple choice	• About EMG For Algebra • The control buttons and its functions • How to play EMG For Algebra • Other guidelines and tips	Quit EMG For Algebra
Subjective	Subjective	Subjective		
Falling bricks	Falling bricks	Falling bricks		
Puzzle	Puzzle	Puzzle		
Exit chapter	Exit chapter	Exit chapter		

Table 5.1: Table of Pyramid Analysis Diagram

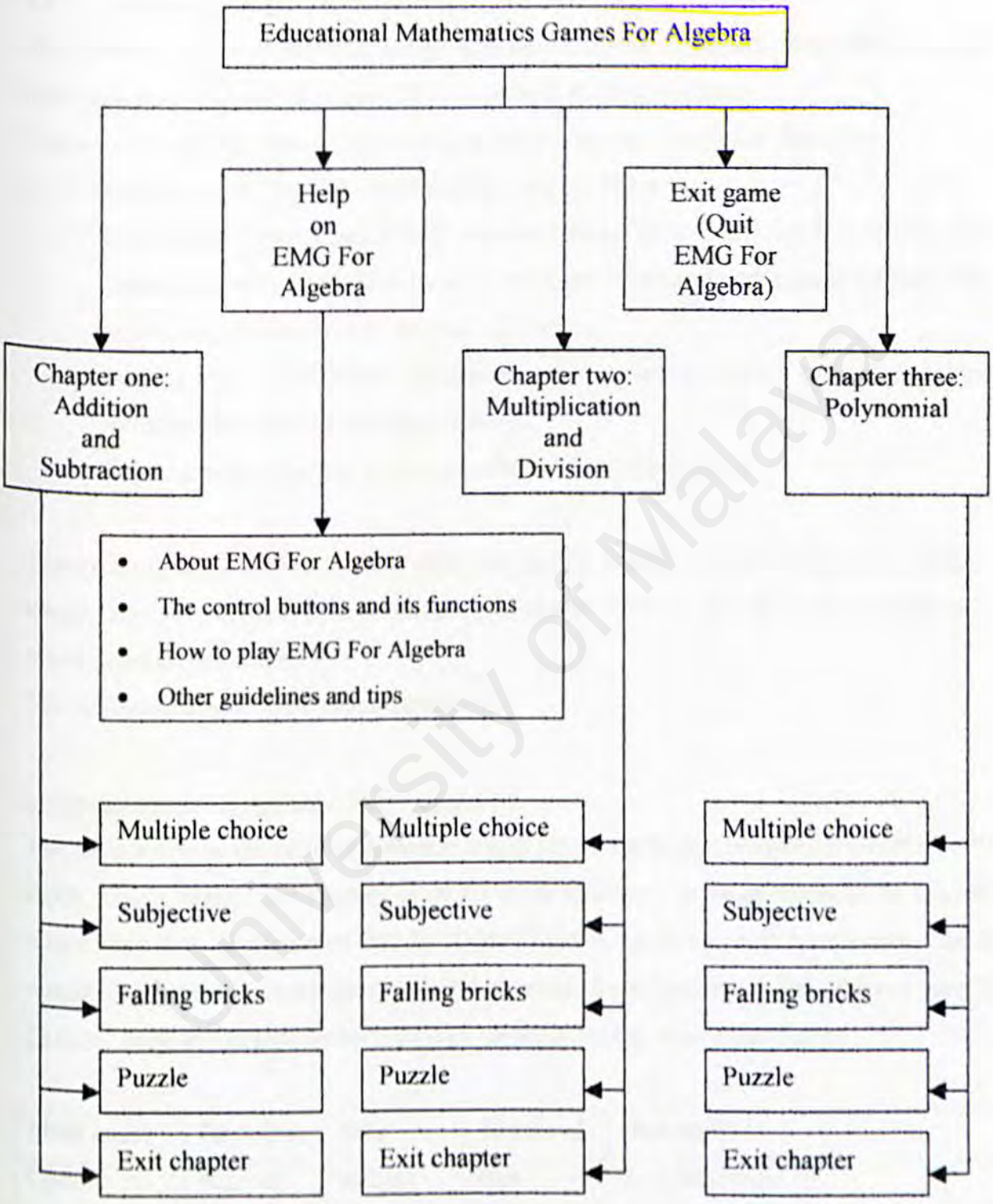


Figure 5.2: Pyramid Analysis Diagram

5.3 Database Design

The primary activity of database design is to select logical representations of data objects (data structures) identified during the requirement definition phase.

There are 4 ways that data is represented at various points in systems' life cycle:

1. External view: The data itself and the context it is in.
2. Conceptual view: English-like representations of external view. Creating the conceptual view from the external view is the process of verbalizing the facts that represented by the data of the external view.
3. Logical view: Represents information as entities, attributes and relationships following the rules of relational theory.
4. Physical view: Physical implementation of the logical view.

The database of EMG For Algebra computer games system consists of four main tables, which cater for multiple choice module, subjective module, falling bricks module and puzzle module respectively.

The following are all the 4 tables mentioned:

Database design for multiple choice module

The table below is the table of database design for multiple choice module. It contains 10 fields. The 'Chapter' field serves as an ID to the questions to assign them to the chapter where they will be presented. While 'Code' field is set to serve as a reference, like a numbering system to those questions and answers. Each question followed by at least 3 choices, some might have up to 5 choices, depends on the level of difficulty.

Field name	Data type	Size	Required	Description
Code	Number	Integer	Yes	As a reference
Chapter	Number	Integer	Yes	Specify the chapter
Question	Text	255	Yes	Mathematical question
Choice1	Text	30	Yes	One of the choices an also the

				correct answer
Choice2	Text	30	Yes	One of the choices
Choice3	Text	30	Yes	One of the choices
Choice4	Text	30	-	Could be one of the choices
Choice5	Text	30	-	Could be one of the choices
Min_level	Number	Integer	Yes	The lowest level that this question can appear
Marks	Number	Integer	Yes	Marks allocated to this question

Table 5.2: Table of Multiple Choice module database design

Database design for subjective module

The table below is the table of database design for subjective module. It contains 6 fields. The time limit field does not apply in this table because in this module, the time limit is not set for the questions individually, but overall for a series of questions in particular level of difficulty. Thus the time limit is set in the program. Within the time limit, player(s) can score maximum 8 questions if they can. The 'Code' field and 'Chapter' field are served the purpose as mentioned in the previous module.

Field name	**Data type**	**Size**	**Required**	**Description**
Code	Number	Integer	Yes	As a reference
Chapter	Number	Integer	Yes	Specify the chapter
Question	Text	255	Yes	Mathematical question
Answer	Text	30	Yes	The correct answer
Min_level	Number	Integer	Yes	The lowest level that this question can appear
Marks	Number	Integer	Yes	Marks allocated to this question

Table 5.3: Table of Subjective module database design

Database design for falling bricks module

Same as the table of database design for subjective module. The table of falling bricks module in the database contains 5 fields.

Field name	Data type	Size	Required	Description
Code	Number	Integer	Yes	As a reference
Chapter	Number	Integer	Yes	Specify the chapter
Question	Text	255	Yes	Mathematical question
Answer	Text	30	Yes	The correct answer
Min_level	Number	Integer	Yes	The lowest level that this question can appear

Table 5.4: Table of Falling Bricks module database design

Database design for puzzle module

This is the largest table in the database design. Here is altogether 75 fields including "Code", "Chapter", "Min_level" and 72 puzzle pieces' information.

Field name	Data type	Size	Required	Description
Code	Number	Byte	Yes	As a reference
Chapter	Number	Integer	Yes	Specify the chapter
P0	Text	50	Yes	One of the pieces in the board
P1	Text	50	Yes	One of the pieces in the board
P2	Text	50	Yes	One of the pieces in the board
P3	Text	50	Yes	One of the pieces in the board

P69	Text	50	Yes	One of the pieces in the board
P70	Text	50	Yes	One of the pieces in the board
P71	Text	50	Yes	One of the pieces in the board
Min_level	Number	Integer	Yes	The lowest level that this question can appear

Table 5.5: Table of Puzzle module database design

All the 4 tables above consist of 5 columns respectively. The first column is to store field name of a particular field, then the 'data type' and 'size' column indicate the data type and the size of the data stored in the field. While the word 'Yes' in the 'Required' column indicates that row must be retrieved when the particular record is retrieved from the database.

5.4 Interface Design

As shown in the pyramid analysis diagram, about 20 interfaces (screen pages) are required for the whole system. Among the main interfaces are main menu interfaces and 3 chapter menu interfaces. Each chapter menu interface is able to call out 5 module interfaces, which are multiple choice, subjective, falling bricks, puzzle and exit chapter module interface. Lastly, there is a 'Help on EMG For Algebra' screen page at the chapter level.

However, the series of module interfaces are almost the same over chapter 1, 2 and 3. Thus, the module level interfaces showing below are only for chapter two. Those for chapter one and three will not be shown because they are quite similar to chapter two. Besides, the chapter menu interfaces will show only for chapter 2, discarding the chapter 1 and 3, due to the same reason.

Graphical User Interface Design For Main Menu

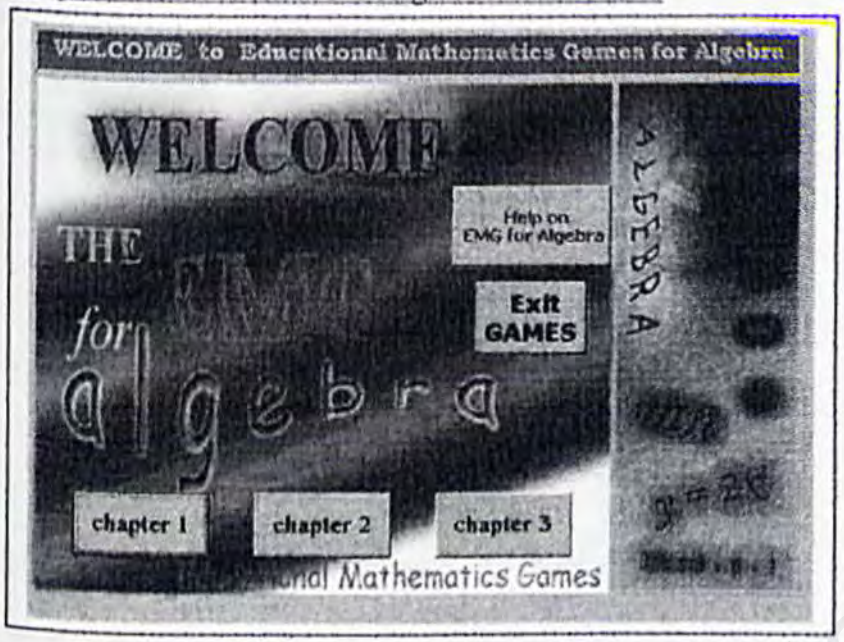


Figure 5.3: Interface design for Main Menu

Graphical User Interface Design For Help On EMG For Algebra

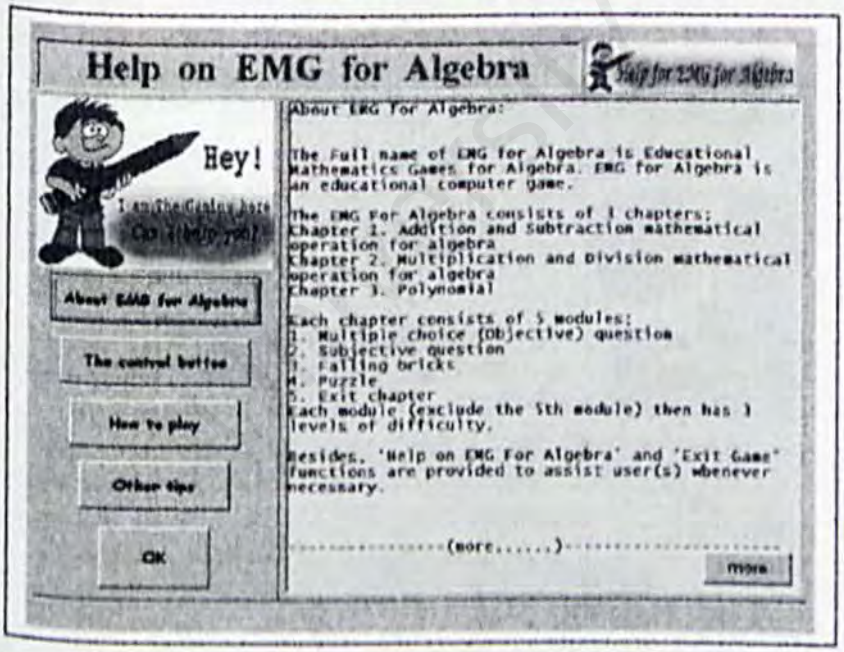


Figure 5.4: Interface design for Help on EMG For Algebra

Graphical User Interface Design For Subjective Module In Chapter Two

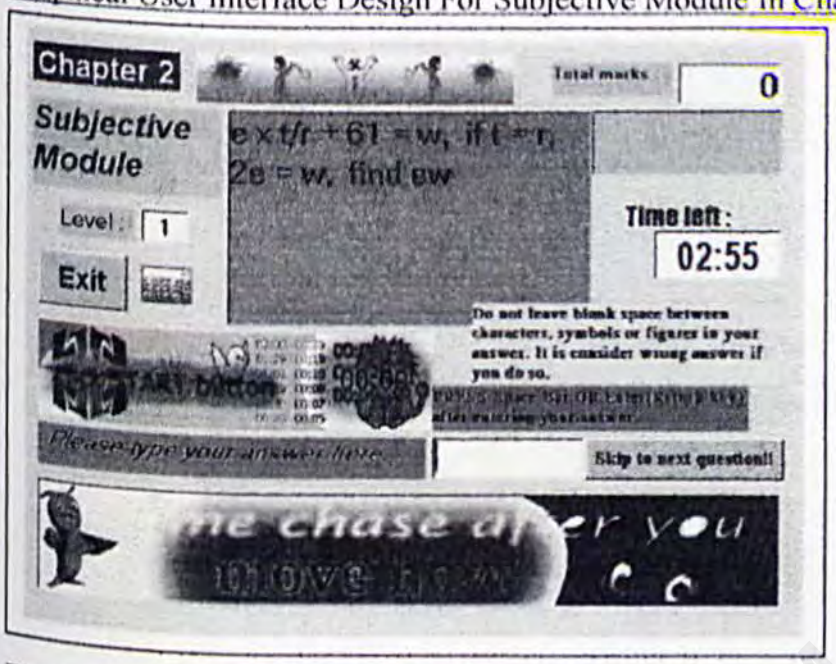


Figure 5.7: Interface design for Subjective module in Chapter Two

Graphical User Interface Design For Falling Bricks Module In Chapter Two

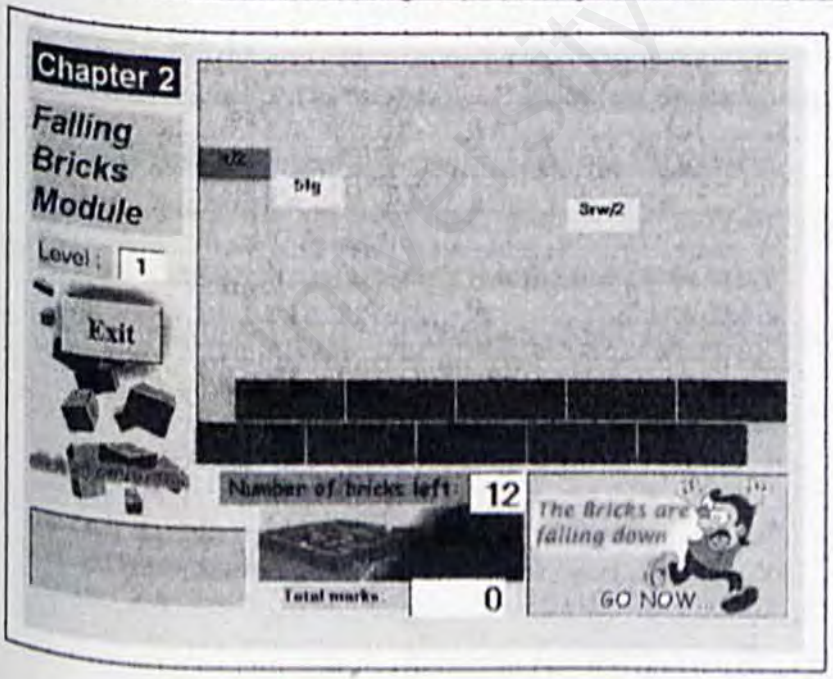


Figure 5.8: Interface design for Falling Bricks module in Chapter Two

Graphical User Interface Design For Puzzle Module In Chapter Two

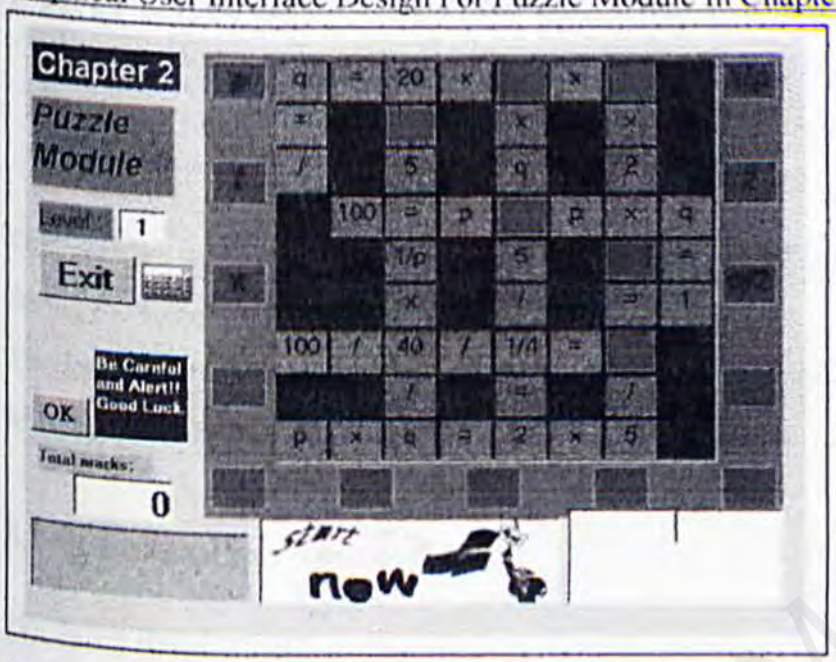


Figure 5.9: Interface design for Puzzle module in Chapter Two

5.5 Project Outcome

At the end of the project, it is expected that the Educational Mathematics Games (EMG) For Algebra will be:-

- Able to help the secondary school students in improving their mathematical problem solving skill through enjoying the games.
- Used by the teachers and parents to motivate their students and children respectively in the learning (mathematics) process.
- Able to included some past year (PMR and SPM examination) questions, to help the students in coping with their examination.

Graphical User Interface Design For Puzzle Module In Chapter Two

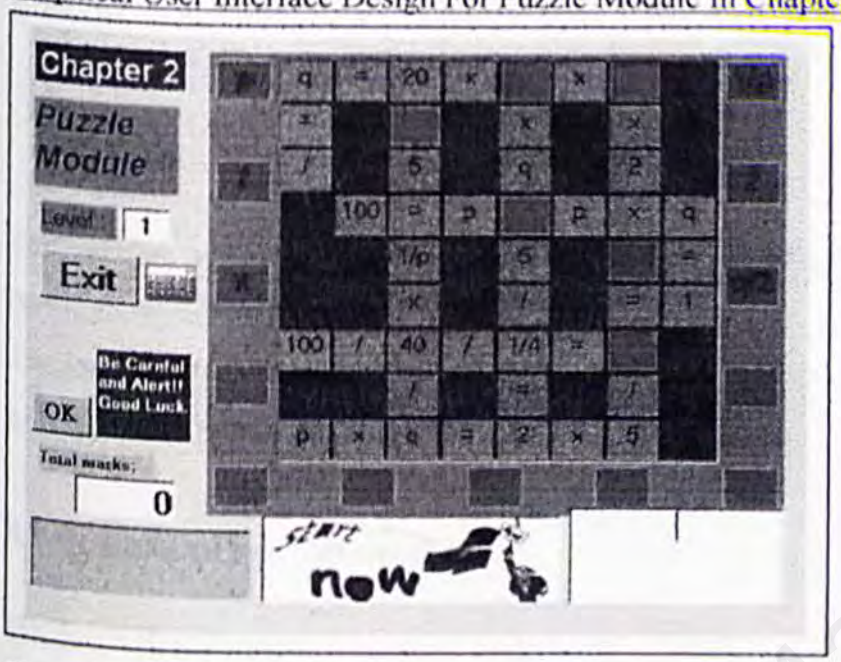


Figure 5.9: Interface design for Puzzle module in Chapter Two

5.5 Project Outcome

At the end of the project, it is expected that the Educational Mathematics Games (EMG) For Algebra will be:-

- Able to help the secondary school students in improving their mathematical problem solving skill through enjoying the games.
- Used by the teachers and parents to motivate their students and children respectively in the learning (mathematics) process.
- Able to included some past year (PMR and SPM examination) questions, to help the students in coping with their examination.

CHAPTER 6: SYSTEM DEVELOPMENT AND IMPLEMENTATION

This section will show some of the minor modifications made to the previous design during the system development process.

The important part of the Visual BASIC 6.0 coding is shown for the purpose of further explanation.

6.1 Development Environment

Hardware Requirement

1. Personal Computer using 500 MHz (or above) processor
2. 128MB of primary memory space
3. 20MB (or more) of available hard disk space
4. Multimedia speaker
5. Keyboard
6. Mouse
7. CD-ROM

Software Requirement

1. Windows 2000 operating system
2. Microsoft Visual Basic 6.0
3. Microsoft Access 2000
4. Adobe Photoshop 6.0
5. MorphStudio Image Editor
6. mp3Trim
7. Mp3 To Wave Converter

6.2 Module Implementation

EMG for Algebra consists of 19 .frm file (form file) and 2 .bas (module file). I am going to show the name of the files and the important part of the Visual BASIC 6.0 coding.

6.2.1 Mainmenu.frm form file

Main menu screen appear when the user launch the game.

' Purpose: User click on the button to visit chapters or 'Help on EMG for Algebra' functions or 'Exit GAMES'

```
Private Sub chapter1_Click()          'triggered when user click 'chapter 1' button
    Image1_Click
    chapter1menu.Show                 'chapter menu of Chapter ONE shown
    Unload Me
End Sub
```

```
Private Sub chapter2_Click()          'triggered when user click 'chapter 2' button
    Image1_Click
    chapter2menu.Show                 'chapter menu of Chapter TWO shown
    Unload Me
End Sub
```

```
Private Sub chapter3_Click()          'triggered when user click 'chapter 2' button
    Image1_Click
    chapter3menu.Show                 'chapter menu of Chapter THREE shown
    Unload Me
End Sub
```

```
Private Sub Exit_Click()              'triggered when user click 'Exit GAMES' button
```

```
    confirmexit.Show                    'the confirm exit dialog box shown
End Sub
```

```
Private Sub help_Click()                'triggered when user click 'Help on EMG for
                                        Algebra' button
    helpEMG.Show                        'Help on EMG for Algebra screen shown
    Unload Me
End Sub
```

6.2.2 Chapter1menu.frm form file

Chapter menu for chapter ONE appear when the user click the 'Chapter 1' button on the main menu screen.

'Purpose: User click on the buttons to choose one of the game modules or exit to the main menu.

```
Private Sub falling_Click()             'triggered when user click 'Falling Bricks' button
chapter1F.Show                          'interface of Falling Bricks module in chapter one
                                        shown
Unload Me
End Sub
```

```
Private Sub multiple_Click()            'triggered when user click 'Multiple Choice' button
chapter1M.Show                          'interface of Multiple Choice module in chapter one
                                        shown
Unload Me
End Sub
```

```
Private Sub puzzle_Click()          'triggered when user click 'Puzzle' button
chapter1P.Show                      'interface of Puzzle module in chapter one shown
Unload Me
End Sub
```

```
Private Sub subjective_Click()      'triggered when user click 'Subjective' button
chapter1S.Show                      'interface of Subjective module in chapter one shown

Unload Me
End Sub
```

```
Private Sub Exit_Click()            'triggered when user click 'Exit Chapter' button
mainmenu.Show                       'exit chapter menu screen for Chapter ONE. Main menu shown

Unload Me
End Sub
```

6.2.3 Chapter1M.frm form file

The interface for multiple choice module in chapter ONE appear when the user click the 'Multiple Choice' button on the chapter menu screen.

'Purpose: User play the multiple choice game.

```
Private Sub reset()                 'retrieve mathematical question from the database and display it on the screen.

    If cnt >= 0 And cnt < 10 Then
```

```
    level.Caption = "1"          'set level of difficulty to 1
    lowcode = 51
    highcode = 100
  End If
  If cnt >= 10 And cnt < 20 Then
    level.Caption = "2"          'set level of difficulty to 2
    lowcode = 1
    highcode = 50
  End If
  If cnt >= 20 And cnt < 30 Then
    level.Caption = "3"          'set level of difficulty to 3
    lowcode = 55
    highcode = 130
  End If

question.DataField = "question"      'set the data field of question box to field
                                     'question' in the table 'Multiple1'
simply                               'randomly pick the question from the
                                     database

If level.Caption = "2" Then          '4 choices of answer in level 2
  LC4.Enabled = True
  LC4_Click
End If
If level.Caption = "3" Then          '5 choices of answer in level 3
  LC4.Enabled = True
  LC5.Enabled = True
  LC4_Click
  LC5_Click
```

```
End If
If level.Caption = "1" Then                    '3 choices of answer in level 1
   LC4.Enabled = False
   LC5.Enabled = False
End If

For t = 1 To 8                                 'set time limit colour clock to function
   If press(cnt) = 0 And level.Caption = "1" Then
      time(t - 1).Visible = True
      Wait 3500
   End If
   If press(cnt) = 0 And level.Caption = "2" Then
      time(t - 1).Visible = True
      Wait 3200
   End If
   If press(cnt) = 0 And level.Caption = "3" Then
      time(t - 1).Visible = True
      Wait 2800
   End If
      marks_Click                              'call marks function
      animation_Change                         'triggers animation function
   End If
  Next t
End If
End Sub
```

```
Private Sub animation_Change()                 'give appropriate animation change
If marka = 1 Then                              'if user select the correct answer
  lTrue.Visible = True
```

```
    Wait 300
End If
If marka = 2 Then                    'if user select the wrong answer
    IFalse.Visible = True
    True3 = 0
    levelT = 0
    Wait 1500
End If
If True3 = 3 Then                    'if user give 3 correct answers in a row
    IThree.Visible = True
    Wait 1500
    True3 = 0
End If
If levelT = 10 Then                  'if user give all correct answers in the
                                     particular level
    ILevel.Visible = True
    Wait 3000
    levelT = 0
End If
End Sub
```

```
Private Sub calculate_Click()        'triggers the calculator facility by clicking
                                     the calculator icon
    calculator.Show                  'the calculator screen shown
End Sub
```

```
Private Sub feedback_Click()         'the feedback screen shown at the end of the
                                     game
    feedback.Visible = True
```

```
fbnumber.Caption = Val(numistake)          'the record of the game shown
fbscore.Caption = Val(marks)

If marks >= 99999 Then                     'if user obtain the highest score =
                                           99999, feedback form shows the
                                           appropriate game record, message
                                           and graphic
   fb4.ForeColor = &HC0C0&
   congrat.Visible = True
   congrat.Caption = "Congratulations!"
   fb3.Caption = "Hey! You have won the game with the . . . . . ."
   fb4.Caption = "  Highest score =          : GOLD key for you!"
   fbscore.Alignment = 1
   fb5.Caption = "You have master in multiple choice module for algebric addition and
subtraction."
   excel.Visible = True
End If

If marks >= 80000 And marks < 99999 Then   'if user obtain the score from 80000
                                           to 99998, feedback form shows the
                                           appropriate game record, message
                                           and graphic
   fb4.ForeColor = &HE0E0E0
   congrat.Visible = True
   congrat.Caption = "Good, Keep it up!"
   fb3.Caption = "You finish the game with the . . . . . ."
   fb4.Caption = "     Score =          : SILVER key for you!"
   fb5.Caption = "You nearly master in multiple choice module for algebric addition
and subtraction."
```

```
    average.Visible = True
  End If

  If marks >= 60000 And marks < 80000 Then      'if user obtain the score from 60000
                                                to 79999, feedback form shows the
                                                appropriate game record, message
                                                and graphic
    fb4.ForeColor = &H80FF&
    congrat.Visible = True
    congrat.Caption = "Not bad, good try!"
    fb3.Caption = "You finish the game with the . . . . . ."
    fb4.Caption = "     Score =          : BROWNZE key for you!"
    fb5.Caption = "You almost master in multiple choice module for algebric addition
and subtraction."
    average.Visible = True
  End If

  If marks >= 40000 And marks < 60000 Then      'if user obtain the score from 40000
                                                to 59999, feedback form shows the
                                                appropriate game record, message
                                                and graphic
    fb4.ForeColor = &HFF0000
    congrat.Visible = True
    congrat.Caption = "No good, watch up!"
    fb3.Caption = "You finish the game with the . . . . . ."
    fb4.Caption = "     Score =          : BLUE key for you!"
    fb5.Caption = "You have not yet master in multiple choice module for algebric
addition and subtraction."
    average.Visible = True
```

```
End If

If marks >= 20000 And marks < 40000 Then        'if user obtain the score from 20000
                                                to 39999, feedback form shows the
                                                appropriate game record, message
                                                and graphic
    fb4.ForeColor = &HC000C0
    congrat.Visible = True
    congrat.Caption = "You've Fail!"
    fb3.Caption = "You finish the game with the . . . . . ."
    fb4.Caption = "     Score =          : PURPLE key for you!"
    fb5.Caption = "You did badly in multiple choice module for algebric addition and
subtraction."
    average.Visible = True
End If

If marks < 20000 Then                           'if user obtain the score less than
                                                20000, feedback form shows the
                                                appropriate game record, message
                                                and graphic
    congrat.Visible = False
    fb3.Caption = "You end up with . . . . . ."
    fb4.Caption = "     Score =          : NO key for you!"
    fb5.Caption = "Why so CHAAA one? You've fail in multiple choice module for
algebric addition and subtraction, DO something!"
    cha.Visible = True
End If
End Sub
```

```
Private Sub marks_Click()                    'add or deduct marks from the total marks
If markc = 1 Then                            'if user make a correct choice
  markc = 3
  marks.Caption = Val(marks.Caption) + Val(mark1)    'marks added to the total marks
      levelT = levelT + 1
      True3 = True3 + 1

If True3 = 3 Then                            'if user make 3 correct choices in a row
  marks.Caption = Val(marks.Caption) + 1555   'bonus marks are added to the total marks
End If

If levelT = 10 Then                          'if user make correct choice for all questions in the particular level
  marks.Caption = Val(marks.Caption) + 10000  'more bonus marks are added to the total marks
End If
End If

If markc = 2 Then                            'if the user make the wrong choice or fail to make any choice before the time limit expired
  markc = 3
  numistake = numistake + 1
  marks.Caption = Val(marks.Caption) - Val(mark1)    'marks are deducted from the total marks
    If marks <= 0 Then
      marks = 0
```

```
    End If
End If
```

```
Private Sub start_Click()
If (start.Caption = "Exit") Then          'if user click 'Exit' button
    chapter1menu.Show                     'quit from the multiple choice module, the chapter menu for chapter ONE shown
    Unload Me
Else
  start.Caption = "Exit"                  'the 'start' button change it's label to 'Exit'
  reset                                   'call the reset function to display new question
End If
End Sub
```

```
Public Sub simply()                       'system randomly pick question from the database
    randq = FormatNumber(Rnd * (highcode - lowcode), 0) + lowcode
    ..........
End Sub
```

6.2.4 Chapter1S.frm form file

The interface for subjective module in chapter ONE appear when the user click the 'Subjective' button on the chapter menu screen.

```
'Purpose: User play the subjective game.

Private Sub reset()                    'retrieve new question from the database
                                       and display it on the screen.
    If cnt >= 0 And cnt < 8 Then
      level.Caption = "1"              'set level of difficulty to level 1
      lowcode = 0
      highcode = 35
      If cnt = 0 Then
        time = 180                     'set time limit to 180 seconds = 3 minutes
      End If
    End If
    If cnt >= 8 And cnt < 16 Then
      level.Caption = "2"              'set level of difficulty to level 2
      lowcode = 36
      highcode = 70
      If cnt = 8 Then
        time = 150                     'set time limit to 150 seconds = 2.5 minutes
      End If
    End If
    If cnt >= 16 And cnt < 24 Then
      level.Caption = "3"              'set level of difficulty to level 2
      lowcode = 71
      highcode = 105
      If cnt = 16 Then
        time = 120                     'set time limit to 120 seconds = 2 minutes
      End If
    End If
```

```
  Simply                                    'randomly pick the question from the
                                            database
  If cnt = 1 Or cnt = 9 Or cnt = 17 Then    'set the time limit digital clock to function
    For tt = 0 To 220
      If time >= 0 And atquit = 2 Then
        timem.Caption = FormatDateTime(time / 1440, vbShortTime)
        Wait 1000
          time = Val(time) - 1
      End If
    Next tt
  End If
End Sub
```

```
Private Sub animation_Change()          'give appropriate animation change
If marka = 1 Then                       'if user submit the correct answer
  lTrue.Visible = True
  Wait 300
End If
If marka = 2 Then                       'if user submit the wrong answer
  lFalse.Visible = True
  True3 = 0
  levelT = 0
  Wait 1800
End If
If True3 = 3 Then                       'if user submit the correct answer for 3
                                        questions in a row
  lThree.Visible = True
  True3 = 0
  Wait 1500
```

```
End If
If levelT = 8 Then                    'if user submit the correct answer for all the
                                      questions in the particular level
  ILevel.Visible = True
  levelT = 0
  Wait 3000
End If
End Sub
```

```
Private Sub marks_Click()             'add or deduct marks from the total marks
If markc = 1 Then                     'if the user submit the correct answer
  markc = 3
  marks.Caption = Val(marks.Caption) + Val(mark1)    'marks are added to the total
                                                     marks
      levelT = levelT + 1
      True3 = True3 + 1
      numq = numq + 1

If True3 = 3 Then                     'if the user submit correct answer for 3
                                      questions in a row
  marks.Caption = Val(marks.Caption) + 2000          'bonus marks are added to
                                                     the total marks
End If

If levelT = 8 Then                    'if the user submit correct answer for all the
                                      questions within the particular level
  marks.Caption = Val(marks.Caption) + 10000         'more bonus marks are added
                                                     to the total marks
End If
```

```
End If
If markc = 2 Then                    'if the user submit the wrong answer
  markc = 3
  marks.Caption = Val(marks.Caption) - Val(mark1)    'marks are deducted from the
                                                     total marks
    If marks <= 0 Then
      marks = 0
    End If
End If
End Sub
```

```
Private Sub skip_Click()             'if the user click 'skip to next question!!'
                                     button
  answer.Text = ""
    reset                            'another question to display
End Sub
```

```
Public Sub simply()                  'system randomly pick question from the
                                     database
  For A = 0 To 23
    randq = FormatNumber(Rnd * (highcode - lowcode), 0) + lowcode
    ..........

End Sub
```

6.2.5 Chapter1F.frm form file

The interface for falling bricks module in chapter ONE appear when the user click the 'Falling Bricks' button on the chapter menu screen.

'Purpose: User play the falling bricks game.

```
Private Sub brick_Click(index As Integer)
  brick(index).BackColor = vbGreen          'set the selected brick to green colour
End Sub
```

```
Private Sub left1_Click()                   'triggered when user press 'z' key
  For X = 0 To 7
    If brick(X).BackColor = vbGreen Then
      brick(X).Move brick(X).Left - 300     'shift the selected brick 300 pixels to
                                            the left side
End Sub
```

```
Private Sub right1_Click()                  'triggered when user press 'c' key
  For X = 0 To 7
    If brick(X).BackColor = vbGreen Then
      brick(X).Move brick(X).Left + 300     'shift the selected brick 300 pixels to
                                            the right side
    End If
  Next X
End Sub
```

```
Private Sub moves_Change()                  ' for the user to move the selected brick
  If Right(moves.Text, 1) = "z" Then        'if user press 'z' key
  left1_Click
```

```
  End If
  If Right(moves.Text, 1) = "c" Then          'if user press 'c' key
  right1_Click
  End If
End Sub
```

```
Private Sub brickfall()          'the bricks will fall automatically when the
                                 game start
For index = 0 To 7
  If brick(index).Visible = True And brick(index).Top < 5710
        And quitauto = 2 And cnt <= 36 Then
    brick(index).Move brick(index).Left + 0, brick(index).Top + 50
                                 'the brick fall in the rate of 50 pixel for
                                 every 40 ms
    Wait 40
  End If
    If brick(index).Top >= 5710 And brick(index).Top < 6000 And quitauto = 2 Then
      brick(index).Top = 5760
      brick(index).BackColor = vbBlue   'the colour of the brick turn blue if
                                        the user fail to match it with any
                                        walls before the brick touch the
                                        bottom part of the wall
    End If
Next index
End Sub
```

6.2.6 Chapter1P.frm form file

The interface for puzzle module in chapter ONE appear when the user click the 'Puzzle' button on the chapter menu screen.

```
'Purpose: User play the puzzle game.

Private Sub OK_Click()                    'User click 'OK' to submit the solved puzzle
                                          or click 'next' to ask for next puzzle board
If (OK.Caption = "Next") Then             'if user click 'next' button
   OK.Caption = "OK"
    Reset                                 'another puzzle board to be solve
Else
  OK.Caption = "Next"                     'if user click 'OK' button
  Marking                                 'call the marking function to mark the
                                          submitted puzzle
  animation_Change
End If
End Sub

Private Sub p_DragDrop(index As Integer, Source As control, X As Single, Y As Single)
  If p(index).BackColor = vbGreen Then    'if user drop a puzzle piece into a
                                          already-filled square
    temp.Caption = p(index).Caption
    p(index).Caption = Source.Caption     'the square now take the value of the
                                          puzzle piece
    Source.Caption = temp.Caption         'the puzzle piece now take the value
                                          of the square
    Source.BackColor = vbGreen            'the colour of the puzzle piece now
                                          turn green
```

```
    End If

    If p(index).BackColor = vbYellow Then        'if user drop a puzzle piece into a
                                                 yellow blank square
        Source.Visible = False                   'the puzzle piece is now disappear
        Source.BackColor = &HFF00FF
        p(index).Caption = Source.Caption        'the blank square now take the value
                                                 of the puzzle piece
        p(index).BackColor = vbGreen             'the yellow square is now turn green
    End If
End Sub
```

The rest of the .frm files, which are not shown here includes chapter2menu.frm, chapter2M.frm, chapter2S.frm, chapter2F.frm, chapter2P.frm, chapter3menu.frm, chapter3M.frm, chapter3S.frm, chapter3F.frm, chapter3P.frm, calculator.frm and confirmexit.frm.

Two of the .bas files, which are also not shown here include delay.bas and audio.bas. The purpose of deley.bas file is to support the wait function used in the .frm file function, while the audio.bas file supports the play sound function used in the .frm file function.

6.3 Documentation

Program documentation is a set of written descriptions that explain to the reader what the program does and how to do it. Internal documentation is a descriptive material written directly within the source code, while all other documentations are regarding as external documentation.

Internal documentation refers to comments within the codes, it provides information that identifies the program, describe its data structures, algorithms and control flow. The header comment block approach is used to provide above-mentioned information. In Visual BASIC 6.0, the comment tag is ', each comment in the VB script must begin with the comment tag '.

External documentation is a part of the overall system documentation. At the time the features were written, much of the rationale for the function structure and flow has already been detailed in the design documents section.

CHAPTER 7: SYSTEM TESTING

System testing and debugging are critical elements in a system development process. The purpose of system testing is to discover bugs and defects that present in the system. A System testing will only be recognized as successful only if errors are detected. Generally, system testing is focusing on:

- Demonstrate that the software has been implemented and has met the software requirement specification.
- Show that errors handling has been done.
- Reveal different classes of errors and introduce the solution for each error.
- Report on any bugs that cannot be resolved. In this case, the programming language bugs.

7.1 EMG for Algebra Under Tested

For the EMG for Algebra computer game, the unit testing, integration testing and system testing has been carried out to test for game's errors. The EMG for Algebra has being tested based on the following generic characteristic:

- The process starts from the main menu, then branches to three chapter menus, and finally the gaming modules for those chapters.
- The highlight of the testing process is the unit test for each module.
- Both White Box testing method and Black Box testing method are used.

7.2 Unit Testing

The highlight of the testing process for EMG for Algebra is the unit test for every gaming module. There are 12 gaming modules to be tested. Besides, there is a 'Help on EMG for Algebra' function to be tested, but it is relatively simple.

The system has undergone different testing techniques which is White Box testing and Black Box testing.

Black Box Testing

Black Box testing is a software testing technique that is done focusing on the software functionality without go through the source code. It is a test case design method that uses the control structure of the procedural design to derive test cases.

The EMG for Algebra has being tested by a few person, they are my friends and my housemates, who has no ideas on what programming is. They just guide by the developer (myself) on how to launch the game and select gaming module by clicking buttons. They will simply select any gaming modules to play and some of them might not follow the normal flow as the developer expect. The testers might start a gaming module and do not know what to do next, here of course the game system will proceed and waiting for the tester to play, or else the system set the tester to lose the game if the tester continue to wait until the time limit expired.

Then the developer will give further instruction to the tester so that he/she can start to play the game. The tester play the game after knowing the aim of the game and the control button used. During the process of this testing, anything can happen, for instance, the tester might suddenly click 'Exit' button to quit the module while the game is going on half way.

This process can check whether the module be able to cope with such sequence of flow, which the system are suppose to support any action that user done without fail. The system should not 'hang' or present wrongly while the tester gives an unexpected mouse click or key press.

Here, I would like to take the falling bricks module to assist my explanation:
The looping function in the falling bricks module is executed at the beginning of the game (after the user press 'start' button) until the game ended. The loop function supposes to end also if the user quit the module half way on the game.
However, the first time when I carried the unit test for falling bricks module, the looping function can't stop even the game ended and cause the system to hang. I detected the bug and solved it. But later I found the system hang again when the user try to quit before the game ended. Therefore, I have to add another logic statement before the looping function to check weather the user still carry on the falling bricks module. If the user already quit the module, the loop should be stopped.
Besides, the user is free to solve or select any brick first and the other one later, the sequence of bricks selected and solve by the user should not bring significant effects to the system. The total marks collected and the bricks falling rate should not affect by the sequence mentioned above.

Black Box testing usually starts after the modules nearly developed and already undergone White Box testing.

White Box Testing

White Box Testing is the technique that exercised by software tester focus on the functional requirement of the software. Here, the software tester will navigate through all the code of the system line by line and performing data debugging in detail.

This technique is actually cost a lot in terms of time consuming. The tester has to understand the effect of every single line of code and the combination of it. Therefore, the tester has to use the debugger to step into the execution of the program. In the unit testing process of EMG for Algebra, the White Box testing technique applied to all

modules. By this way, the tester can trace out the portions of program, which do not work as expected.

White box testing attempt to find errors on the data validation in the system. Besides, it wills also checkout all the logical decision in detail. The tester is debugging based on the flow of the game and system's respond to the user's input. The testing and coding was repeated again and again until the expected result obtained.

Here, I would like to take the puzzle module for example:

When user playing puzzle module, there are few possibility can happen,

1. User places the correct puzzle piece into the yellow blank square
2. User places the wrong puzzle piece into the yellow blank square
3. User replaces another puzzle piece (which is the correct puzzle piece) into the already-filled square
4. User replaces another puzzle piece (which is the wrong puzzle piece) into the already-filled square
5. User drops the puzzle piece outside the puzzle board.
6. User does not place any puzzle piece into the yellow blank square.

Well, following the above situation, the system respond should be:

1. Marks are added to the total marks if the user clicks 'OK' button to submit the puzzle board.
2. Marks are deducted from the total marks if the user clicks 'OK' button to submit the puzzle board.
3. Marks are added to the total marks if the user clicks 'OK' button to submit the puzzle board.
4. Marks are deducted from the total marks if the user clicks 'OK' button to submit the puzzle board.
5. The puzzle piece automatically placed back to its initial location, system sounds a warning tone to the user.

6. It is consider user place a wrong puzzle piece to the blank square. Marks are deducted from the total marks if the user clicks 'OK' button to submit the puzzle board.

Ok, that is what I expected the outcome of the system before debugging. But I found the system respond such a way:
Marks were not deducted when user does not place any puzzle piece into the blank square. Next, when the user drops the puzzle piece outside the board, the puzzle piece disappears.
Consequent of this, I have to step into the code line by line to find out the missing loop and do the correction there.

White Box testing is generally focus on source code of the system. As the result, tester will more concentrate to check the missing functions or invalid function that occurs in the system. Besides, redundant functions were list out and will be removing from the system to reduce the complexity of the code.

7.3 Integration Testing

In short, Integration Testing is a process to verify that all the elements that compose the product can run together. Integration testing was carried out for EMG for Algebra games system to ensure that the sequence of chapter menu calling and gaming module calling are always work consistently.
For the case of EMG for Algebra, the integration testing process is the simple one. Since the modules in the system are nearly independent from each other, the testing cycle does not focus on this part.

Integration testing for EMG for Algebra is focusing on two aspects,

1. User can travel from the higher hierarchical interface (main menu) to second higher hierarchical interface (chapter menu, 'help on EMG for Algebra' function) and to the lowest hierarchical interface (gaming modules). Vice versa, user shall be able to return to the higher hierarchical interface by clicking those exit buttons.
2. The execution of each modules shall not affect the others, all modules' execution are independent from each other.

There are two approaches for exercising the integration testing. It is call incremental integration strategy, the bottom-up integration and regression testing approach. And for EMG for Algebra, bottom-up testing has need used. The main reason that this approach being chosen was because simple to implement.

For the first aspect of EMG for Algebra integration testing focus, the tester will check the system based on the algorithm:

1. User double-clicks 'EMGforAlgebra' icon on the desktop: main menu screen appears.

2. User clicks,

   2.1 'chapter 1' button: chapter menu Chapter ONE appears

   User clicks,

   2.1.1 'Multiple Choice' button: Multiple Choice Module gaming interface appears

   User clicks,

   2.1.1.1 'Start' button: Multiple choice game starts

   2.1.1.2 'Exit' button: game ends, return to chapter menu Chapter ONE

   2.1.2 'Subjective' button: Subjective Module gaming interface appears.

   User clicks,

2.1.2.1 'Start' button: Subjective game starts

2.1.2.2 'Exit' button: game ends, return to chapter menu Chapter ONE

2.1.3 'Falling Bricks' button: Falling Bricks Module gaming interface appears

User clicks,

2.1.3.1 'Start' button: Falling bricks game starts

2.1.3.2 'Exit' button: game ends, return to chapter menu Chapter ONE

2.1.4 'Puzzle' button: Puzzle Module gaming interface appears

User clicks,

2.1.4.1 'Start' button: Puzzle game starts

2.1.4.2 'Exit' button: game ends, return to chapter menu Chapter ONE

2.1.5 'Exit Chapter' button: Return to main menu screen

2.2 'chapter 2' button: chapter menu Chapter TWO appears

User clicks,

2.2.1 'Multiple Choice' button: Multiple Choice Module gaming interface appears

User clicks,

2.2.1.1 'Start' button: Multiple choice game starts

2.2.1.2 'Exit' button: game ends, return to chapter menu Chapter TWO

2.2.2 'Subjective' button: Subjective Module gaming interface appears.

User clicks,

2.2.2.1 'Start' button: Subjective game starts

2.2.2.2 'Exit' button: game ends, return to chapter menu Chapter TWO

2.2.3 'Falling Bricks' button: Falling Bricks Module gaming interface appears

User clicks,

2.2.3.1 'Start' button: Falling bricks game starts

2.2.3.2 'Exit' button: game ends, return to chapter menu Chapter TWO

2.2.4 'Puzzle' button: Puzzle Module gaming interface appears

User clicks,

2.2.4.1 'Start' button: Puzzle game starts

2.2.4.2 'Exit' button: game ends, return to chapter menu Chapter TWO

2.2.5 'Exit Chapter' button: Return to main menu screen

2.3 'chapter 3' button: chapter menu Chapter THREE appears

User clicks,

2.3.1 'Multiple Choice' button: Multiple Choice Module gaming interface appears

User clicks,

2.3.1.1 'Start' button: Multiple choice game starts

2.3.1.2 'Exit' button: game ends, return to chapter menu Chapter THREE

2.3.2 'Subjective' button: Subjective Module gaming interface appears.

User clicks,

2.3.2.1 'Start' button: Subjective game starts

2.3.2.2 'Exit' button: game ends, return to chapter menu Chapter THREE

2.3.3 'Falling Bricks' button: Falling Bricks Module gaming interface appears

User clicks,

2.3.3.1 'Start' button: Falling bricks game starts

2.3.3.2 'Exit' button: game ends, return to chapter menu Chapter THREE

2.3.4 'Puzzle' button: Puzzle Module gaming interface appears

User clicks,

2.3.4.1 'Start' button: Puzzle game starts

2.3.4.2 'Exit' button: game ends, return to chapter menu Chapter THREE

2.3.5 'Exit Chapter' button: Return to main menu screen

2.4 'Help on EMG for Algebra' button: Help on EMG for Algebra screen appears.

User clicks,

2.4.1 'OK' button: Return to main menu screen

2.5 'Exit GAMES' button: reconfimexit dialog box appears

User clicks,

2.5.1 'Yes, I quit now' button: system quits

2.5.2 'No, bring me back' button: return to the main menu screen

For the second aspect of EMG for Algebra integration testing focus, the tester will play around all gaming module again and again without quit the system, the system respond must consistent each time the tester playing the same gaming module

7.4 System Testing

System testing is a series of different tests whose primary purpose is to fully exercise the computer-based system.

EMG for Algebra has undergone system testing focused on a few characteristic as shown below:

- Functional testing

The purpose of functional testing is to ensure the system fulfill functional requirements. The game is played by some invited tester who has no idea on computer programming and do not know what is happening at the development side. The system is suppose to:

1. Display proper mathematical question according to which gaming module that the tester is currently in.
2. Add marks and show appropriate message whenever user give a correct answer.
3. Deduct marks, show the correct answer and appropriate message whenever user give the wrong answer.

- Performance testing

EMG for Algebra is designed to respond fast and accurate to the user's input. The system has to include graphics, audios and animations to fulfill the multimedia specification. The message to acknowledge the user about his/her current performance has to be concise and precise, but the correction respond to the user's wrong answer has to pause for a while.

- Stability testing

The system also tested for its stability, which the system was left by itself without user input for a long period of time. The system then proceeds by counting the time limit and assumes user gives the wrong answer when the time limit expired. The gaming module then show the score = 0 at the end of the game and back to the chapter menu, waiting for next step from the user. The system is found still ready for use when the tester come back to click the buttons on the chapter menu screen.

7.5 Database Testing

The database testing for EMG for Algebra also very simple. The database for EMG for Algebra is a 'open database', although the database is read only from the system point of view, the user actually can open the database table and do the modification on the records.

The tester open the database table and add mathematical questions as well as the answer to the table, then the tester save the modification and launch the EMG for Algebra to test the database. The newly added questions have the equal chance to be picked as the displayed question on the screen.

During the database testing, the tester also has to make sure that the question retrieved does not display at the wrong chapter or wrong module or wrong level of difficulty.

CHAPTER 8: SYSTEM EVALUATION AND CONCLUSION

8.1 Project Objective Achieved

The main objectives of developing this Educational Mathematics Games (EMG) For Algebra are as below:

1. To enable the players enjoy themselves throughout the learning process by playing EMG For Algebra computer games. The algebra topic will no longer be bored and dull but full of fun and challenging.

This objective achieve through putting the mathematical questions in the exercise books appear live in the form of computer game. A game to play is certainly more fun and challenging compare to the homework to do.

2. To train the players think fast and have confident with their own answers. The players are much encouraged to come out with their own 'shortcut' way (any other alternative way) of solution methods, rather than just memorize the textbooks all the time.

This objective achieve through the implementation of time limit concept in the system The time limit concept appears to 'force' the players to find their way out in order to obtain higher score. The players have to bear in mind that they can never obtain better score if they do not use their understanding of algebra concept to answer the questions.

3. To implant the latest technology (computer aided and multimedia) into the mathematics learning process. To enable players learn on their own rate and up to their full potential.

The players will now feel answering the mathematical questions is not to pleasure their teachers or parents, but for their own benefits. They learn while playing the game at their own pace without interference from the others.

8.2 System Strengths

Below are the strengths of EMG for Algebra computer game system that contribute in achieving the objectives stated above.

Easy to Launch

User double clicks on the 'EMGforAlgebra' icon to launch the game. There is no need to remember any password or user ID.

Simple and User-Friendly Interface

The Graphical User Interface in EMG for Algebra is easy to understand and user-friendly. The interface designed to suit a wide spectrum of user, especially to the secondary school students. The user should be able to play the game even without reading the user manual. What the first-time visitor needs to do is to trouble one of his/her friends (who had already knew how to play the game) for less than 5 minutes. Otherwise the first-time visitor can just simply click on the buttons to find his/her own way. The user can also read the help text in the 'Help on EMG for Algebra' function. The help function button is placed on the main menu screen.

Effective Error Recovery

EMG for Algebra is a read only system. The word 'read only' here means, the user cannot input any data or records into the database. Thus, when we talk about the system error recovery, we already ruled out the possibility of system fault caused by invalid data inserted into the database. The only reason of system error for this system is the user clicks the wrong button or presses the wrong key. However, no matter which key or button that the user wrongly presses, it is recoverable. For example, if the user wrongly clicks the 'Puzzle' button instead of 'Exit Chapter' button when the user actually intend to return to the main menu. Then the user can easily quit the Puzzle Module and then click on the 'Exit Chapter' button as he/she wish.

System Available

Since EMG for Algebra is a stand-alone system, it is always available whenever the PC is on. The database of this system is read only and thus ruled out the problem caused by the database locked by another module or user.

Easy to Do Database Maintenance

Users are able to do housekeeping for database maintenance. They can create, add, modify, update and delete the mathematical questions and test it with the game system. The users are allowed to open the database and update the data inside. Each record (the mathematical question) in the database is independent from each other.

Able To Edit The Mathematical Questions To Suit The Standards of Different Users.

The teachers or parents or anybody who can read the user manual, be able to edit the mathematical questions in the database, according to the standard of their students or children.

For example, there is a Subjective question in chapter ONE:

rp - xy = 19, if xy = 11, 4rp - 3xy + 9 =

Well, this question is actually designed for the secondary school student, if the parents want to modify the question to suit their children which in primary school, they can change the question to the following form:

A - 11 = 19, 4A – 33 + 9 =

Thus, the question becomes easier.

Feedback Screen

The game system shows a feedback screen at the end of every gaming module. The feedback screen shows:

1. The number of questions that the user has answered.
2. The number of mistakes that the user has done.
3. The total marks (score) that the user obtains.
4. The key award that the user get.
5. Appropriate message shown.
6. Appropriate graphic shown.

In short, the feedback screen will let the user knows about his/her achievement in the particular gaming module. The feedback screen design is in such a way to congratulate those who has master the particular gaming module, to motivate those who has not, and to give some advice for those who did badly in the game.

Fast Respond to User Input

This is actually a very important requirement for all computer games. EMG for Algebra response fast to every user input. From the screen loading point of view, every menu

screen and gaming module interface can be loaded within 2 seconds after the user's action.

The system responses even faster when the game started. For example, when the user press Enter key to submit the answer for a subjective question, the system be able to decide weather the submitted answer is correct/wrong, then marks should be added/deducted, show appropriate animation and show the correction of the answer (if user submit the wrong answer) within half a second. Therefore, in the user point of view, the system is said to response immediately after user input.

8.3 System Limitations

Due to project boundaries, there are some limitations in EMG for Algebra computer game system. The limitations are as stated below:

Lack of Detail Research and Prototyping Work

The EMG for Algebra system development project has to be finished within 2 semesters, therefore it is quite impossible for me to go in very detail research and prototyping work when developing the games.

The sequence of lack of research and prototyping work is the developer never knows what the majority wants the system to be.

Covers Only Part of Secondary School Syllabus

This game covers only basic part of the world of algebra. EMG for Algebra failed to cover the whole secondary school syllabus but just portion of it. Therefore, the teachers and the secondary school students cannot totally replace their exercise books with this game.

Although the pre-university student and college student are listed as the system's target user, the questions in the database actually do not cater for their standard, mostly

questions are too simple for them. However, they can try on EMG for Algebra as to revise their simple algebra problem solving skill.

Limited Modules

Only four gaming modules for each chapter, which are multiple choice module, subjective module, falling bricks module and puzzle module. For those youngsters who are 'computer game expert' might get bored if they take EMG for Algebra as an usual arcade game, instead of an educational game.

Limited Number of Question

Due to the time limit factor, there are limited variety of questions in the databases. The user could have getting bored after playing many rounds in the same module because the questions keep on repeating.

8.4 Problems Encountered and Solutions

Difficulties In Choosing A Development Technology, Programming Language and Tools

There are many software tools available to develop a software game system currently as stated in the earlier chapters. Choosing a suitable technology and tool was a critical process as all tools have their strengths and weaknesses. In addition, the availability of the required tools for development is also a major consideration. A tough decision arises in choosing between JAVA and Visual BASIC to develop my system.

Solutions:

Advises and views were sought from project supervisor, course mates and even seniors engaging in similar project. Furthermore, surfing the Internet and visiting the document room helped to clarify some doubts.

Determining Scope of The System

When I was about to proposed my system, I faced a tough time to decide the scope of my project. I really couldn't make up my mind, weather to propose a huge system or to scale down my project. I also wondering I should focus on the algebra or geometry or else aspect.

Solutions:

I surf the Internet to visit more mathematical game sites. From there I get some ideas on how an educational mathematics game shall look like. Then, I consulted my project supervisor, course mates and seniors. Finally, I decided on the scope of my project with the advices from my supervisor.

Inexperience In The Chosen Programming Language

Since I has no prior knowledge of programming in Visual BASIC, there was an uncertainty on how to organize the codes and graphical user interface by using VB language. I had only the duration of less than 3 months to pick up VB language do the coding for the entire system. I got a bit miserable on how to start the hard coding, and which module to develop first?

Solutions:

I borrowed a book from my friend to study Visual BASIC 6.0. The book is Programming in Visual BASIC 6.0 by Julia Case Bradley and Anita C. Millspaugh. This book has became my handbook to assist me throughout the coding process. Besides, I downloaded some sample source code from the Internet for my reference purpose. The MSDN library also helps me a lot in my coding process.

8.5 Future Enhancement

Future enhancement can be done to make the system more attractive and cater for more users.

- **Updating The Database**

More mathematical questions can be added to the database, so the question prompt up to the user will not keep on repeating. The questions already in the database can be modified to suit different level of users. For instance, the primary kids or for the undergraduates.

- **Attractive Buttons**

The buttons interface can be modified to become more attractive. The appearance of the buttons can be customized for EMG for Algebra to replace the current uniform rectangular buttons.

- **Attractive Animations and Sounds**

More graphic movements, cartoons and sounds can be edited and inserted into the system.

- **More Gaming Modules**

More gaming modules can be added to the EMG for Algebra. For example, mathematical chess module, treasure hunt module or other multiplayer gaming modules can be added to make the game more interesting.

- **Attractive Help Screen**

More features such as text animations, graphic movements and games demonstration can be implemented to the 'Help on EMG for Algebra' facility.

- **Interface to Monitor The Database**

A database monitoring interface can be created for the purpose of updating database. The user needs not to exit the system in order to modify the database.

- **Extend to Other Aspect of Educational Game**

The concept of EMG for Algebra can be extended to more educational aspects. The system can be modified to come out with EMG for Geometry, EMG for Chemistry and so on.

- **Keep user record**

One more feature can be added to the system to keep user score each time after user plays the game. This is to enable the user to check his/her achievement compare to the previous one.

8.6 Conclusion

EMG for Algebra system has achieved and fulfilled the objectives and requirements as an educational computer game system as specified during system analysis. The students, teachers, parents and anybody else who wish to use the system can do so by installing the application into their personal computer. EMG for Algebra can be regarded as a new teaching aid to help those who easily get bored with their exercise books.

I learn a lot throughout the process of doing this project. This includes knowledge in Visual BASIC programming technique, monitoring database, implementing audio, graphics and animations into the system and so on.

After finished developing the system, I began to realize that even though programming skills and techniques are important in development, good software engineering techniques must also be applied. Here, theories and knowledge gained throughout the course of computer science studies like system analysis, design and software engineering were literally put into practice.

Finally, there are much more rooms for improvement in my system as I stated down in the section of 'Further Enhancement'. I hope that, more and more made in Malaysia educational software can emerge in the future for the benefit of our younger generations, by the way I will continue to work on this, hopefully.

REFERENCES

1. David M. Kroenke, DATABASE PROCESSING: Fundamentals, design, and Implementation, 6th edition, Prentice Hall, Inc. 1998.

2. William Stallings, Ph. D., Richard Van Slyke, Ph. D., Business Data Communications, 3rd edition, Prentice Hall, Upper Saddle River, New Jersey 07458.

3. Rodina Ahmad, Zarinah Mohd. Kasirun, Fariza Hanum Nasaruddin, Pengaturcaraan C++, 2000.

4. Ida M. Flymn and Ann Melver Me Hoes, Understanding Operating System, 2nd edition, International Thomson Publishing, 1997.

5. Abdullah Embong, Ph. D., Sistem Pangkalan Data: Konsep asas, Rekabentuk dan Pelaksanaan, Tradisi Ilmu Sdn. Bhd., 2000.

6. Harvey M. Deitel, Paul J. Deitel, C How To Program, Prentice Hall, Englewood Cliffs, New Jersey 07632.

7. Grant Palmer, Java Programmer's Reference, Wrox Press Ltd., United Kingdom, 2000.

8. The Eng Kiat, Ooi Yong Seang, Yong Kuan Yeoh, FOKUS JINGGA Matematik PMR, Penerbitan Pelangi Sdn. Bhd., Selangor Darul Ehsan, 2000.

9. OO. Java Berorientasikan Objek Menggunakan Java, 2001.

10. Java How To Program, 3rd edition, 1999.

11. Michael Eraut, Fundamental of Intermediate Algebra, McGraw-Hill Book Company, U.S.A., 1970.

12. Robert G. Moon, Fullerton College, Understanding Elementary Algebra, Charles E. Merrill Publishing Company, U.S.A., 1978.

13. Travers, Dalton, Brunner, Using Algebra, Third Edition, Laidlaw Brothers Publishers, U.S.A., 1984.

14. Raymond A. Barnett, Intermediate Algebra Structure and Use, McGraw-Hill Book Company, U.S.A., 1971.

15. John H. Minnick, Raymond C. Strauss, Beginning Algebra, 2nd Edition, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.